

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇOK AMAÇLI ÇİZELGELEME PROBLEMİNİN
GENETİK-MULTIMOORA HİBRİT ALGORİTMASI
İLE ÇÖZÜMÜ**

YÜKSEK LİSANS TEZİ

Mine Büşra GELEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Tez Danışmanı : Dr. Öğr. Üyesi Alparslan Serhat DEMİR

Mayıs 2018

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK AMAÇLI ÇİZELGELEME PROBLEMİNİN
GENETİK-MULTIMOORA HİBRİT ALGORİTMASI
İLE ÇÖZÜMÜ**

YÜKSEK LİSANS TEZİ

Mine Büşra GELEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 31.05.2018 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

**Doç. Dr.
Gülşen AKMAN
Jüri Başkanı**



**Dr. Öğr. Üyesi
Alparslan Serhat DEMİR
Üye**



**Dr. Öğr. Üyesi
Ali GÜLBAĞ
Üye**



BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Mine Büşra GELEN

31.05.2018

TEŞEKKÜR

Yüksek lisans eğitimim süresince fikirlerini, bilgi ve deneyimlerini benimle paylaşan, çalışmamın başlangıcından itibaren her adımında destek olan, teşvik eden, beni yönlendiren sayın danışman hocam Dr. Öğr. Üyesi Alparslan Serhat Demir’e teşekkürlerimi sunarım.

Yazılım bilgisi ile çalışmamın yazılım sürecinde katkı sağlayan, Bilgisayar Öğretmeni Sayın Umut Ali Yurdakul’a katkılarından dolayı teşekkür ederim.

Bu çalışmamda ve hayatım boyunca yaptığım tüm çalışmalarımda beni destekleyen, anlayışını, sabrını ve sevgisini esirgemeyen aileme teşekkürlerimi sunarım.

Bu çalışma SAÜ Bilimsel Araştırma Projeleri Komisyonu tarafından desteklenmiştir. (Proje No: 2017-50-01-073) Desteğinden dolayı SAÜ Bilimsel Araştırma Projeleri Komisyonu’na teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	vi
TABLOLAR LİSTESİ.....	ix
ÖZET.....	x
SUMMARY	xi
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
LİTERATÜR ÖZETİ.....	3
2.1. Akış Tipi Çizelgeleme Problemleri	3
2.2. Genetik Algoritmalar ve Çok Kriterli Karar Verme Yöntemleri	7
BÖLÜM 3.	
ÇİZELGELEME	8
3.1. Atölye Tipi Çizelgeleme	9
3.2. Akış Tipi Çizelgeleme.....	10
BÖLÜM 4.	
METOTLAR.....	12
4.1. Genetik Algoritmalar.....	12
4.1.1. Seçim.....	14
4.1.2. Çaprazlama.....	16

4.1.3. Mutasyon.....	18
4.1.4. Eleme.....	19
4.2. MultiMoora Metodu.....	20
4.2.1. Moora-oran metodu.....	20
4.2.2. Referans noktası yaklaşımı.....	22
4.2.3. Tam çarpım formu.....	22
4.2.4. Sıra baskınlık teorisi.....	23
 BÖLÜM 5.	
UYGULAMA	25
5.1. Önerilen Hibrit Modeller.....	25
5.1.1. Genetik algoritmaların seçim adımında MultiMoora metodunun uygulanması: MultiMoora tabanlı genetik algoritma-1 (MBGA-1).....	25
5.1.2. Genetik algoritmaların eleme adımında MultiMoora metodunun uygulanması: MBGA-2 hibrit algoritması.....	34
5.1.3. Genetik algoritmaların seçim ve eleme adımında MultiMoora metodunun uygulanması: MBGA-3 hibrit algoritması.....	36
5.2. Problemin Yapısı.....	39
5.3. Algoritmaların Program Üzerinde Uygulanması	40
 BÖLÜM 6.	
SONUÇ VE DEĞERLENDİRME.....	42
 KAYNAKLAR	71
EKLER.....	76
ÖZGEÇMİŞ	83

SİMGELER VE KISALTMALAR LİSTESİ

y_j^*	: j alternatifinin tüm kriterlere göre normalize edilmiş değeri
A_j	: Her bir alternatif için maksimum sütun değerleri çarpımı
A_m	: m . alternatif
B_j	: Her bir alternatif için minimum sütun değerleri çarpımı
C_{max}	: Maksimum tamamlanma zamanı
C_n	: n . kriter
\bar{F}	: Ortalama akış süresi
\bar{T}	: Ortalama gecikme
T_{max}	: Maksimum gecikme
U_j	: Alternatif j 'nin genel faydası
d_i	: i . işin teslim tarihi
r_i	: Referans değeri
t_{ij}	: i . işin j . tezgahdaki işlem süresi
w_i	: i . kriter ağırlığı
x_{ij}	: j . alternatifinin i . kriter açısından performans ölçüm değeri
AHP	: Analitik Hiyerarşi Prosesi
HQGA	: Hibrit kuantumdan esinlenmiş bir genetik algoritma
L-NSGA	: Lorenz dominantlık ilişkisini kullanan Dominant Olmayan Sıralama Genetik Algoritması
MBGA	: MultiMoora tabanlı genetik algoritma
MOGA	: Çok amaçlı genetik algoritma
MOORA	: The multi-objective optimization on the basis of ratio analysis
MOPS	: Çok objektif bir parçacık sürüsü
MOPSO	: Hibrit çok objektif parçacık sürüsü optimizasyonu
MultiMoora	: Moora plus the full multiplicative form

N	: Popölasyon büyüklüğü
NEH	: Nawaz, Enscore ve Ham tarafından geliştirilen sezgisel bir yöntem
NP-Zor	: Belirsiz olmayan polinom zamanı
NSGA	: Dominant Olmayan Sıralama Genetik Algoritması
PGA-ALS	: Pareto Genetik Algoritması- yerel olmayan bir araştırmaya tabi olan dominant olmayan çözümlerin arşivi
SGA	: Simpleks büyüme algoritması
VEGA	: Vektör Değerli Genetik Algoritma
NLJ	: Geciken iş sayısı
NTJ	: Zamanında biten iş sayısı
$f(x)$: Uygunluk fonksiyonu
$f(x)_{ort}$: Ortalama uygunluk fonksiyonu
$f(x)_{min}$: Minimum uygunluk fonksiyonu
g	: Maksimize edilecek hedeflerin sayısı
is	: İş sayısı
m	: MBGA algoritmalarında belirli bir işlemi uygularken seçilen birey sayısı
ms	: Tezgah sayısı
n	: Hedeflerin sayısı
α	: Alfa
β	: Beta
γ	: Gama
δ	: Delta
θ	: Teta

ŞEKİLLER LİSTESİ

Şekil 3.1. Atölye tipi çizelgeleme	10
Şekil 3.2. Akış tipi çizelgeleme	11
Şekil 4.1. 15 işten oluşan çok amaçlı akış tipi çizelgeleme problemi için kromozom örneği	13
Şekil 4.2. 15 işin çizelgelendiği örnek bir kromozom üzerinde doğrusal sıralı çaprazlamanın farklı bir uygulaması.....	17
Şekil 4.3. 15 işin çizelgelendiği örnek bir kromozoma keyfi üç geni değiştirme mutasyonu uygulanması	18
Şekil 5.1. Önerilen MBGA-1-A hibrit algoritması akış şeması	28
Şekil 5.2. Önerilen MBGA-1-B hibrit algoritması akış şeması	31
Şekil 5.3. Önerilen MBGA-1-C hibrit algoritması akış şeması	33
Şekil 5.4. Önerilen MBGA-2 hibrit algoritması akış şeması	35
Şekil 5.5. Önerilen MBGA-3 hibrit algoritması akış şeması	38
Şekil 5.6. Algoritmaların çok amaçlı akış tipi çizelgeleme probleminde uyulandığı programa ait ekran görüntüsü	41
Şekil 6.1. 15 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	43
Şekil 6.2. 25 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	44
Şekil 6.3. 50 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	45
Şekil 6.4. 100 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	46
Şekil 6.5. 15 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği	46

Şekil 6.6. 25 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	47
Şekil 6.7. 50 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	48
Şekil 6.8. 100 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	49
Şekil 6.9. 15 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	49
Şekil 6.10. 25 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	50
Şekil 6.11. 50 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	51
Şekil 6.12. 100 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	51
Şekil 6.13. 15 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	52
Şekil 6.14. 25 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	53
Şekil 6.15. 50 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	53
Şekil 6.16. 100 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	54
Şekil 6.17. 15 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	55
Şekil 6.18. 25 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	55
Şekil 6.19. 50 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	56
Şekil 6.20. 100 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200	
jenerasyon $f(x)$ ort değerleri değişimi grafiği.....	57
Şekil 6.21. 15 iş test problemi MBGA-3 algoritması ve MOGA'nın 200	
jenerasyon $f(x)$ min değerleri değişimi grafiği	57

Şekil 6.22. 25 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği	58
Şekil 6.23. 50 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği	59
Şekil 6.24. 100 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği	59
Şekil 6.25. 15 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği	60
Şekil 6.26. 25 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği	61
Şekil 6.27. 50 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği	62
Şekil 6.28. 100 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ ort değerleri değişimi grafiği	63
Şekil 6.29. 15 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği.....	63
Şekil 6.30. 25 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği.....	64
Şekil 6.31. 50 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği.....	65
Şekil 6.32. 100 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)$ min değerleri değişimi grafiği.....	66

TABLÖLAR LİSTESİ

Tablo 6.1. 15 iş test problemi üzerinde algoritmaların $f(x)$ min değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu.....	67
Tablo 6.2. 25 iş test problemi üzerinde algoritmaların $f(x)$ min değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu.....	67
Tablo 6.3. 50 iş test problemi üzerinde algoritmaların $f(x)$ min değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu.....	68
Tablo 6.4. 100 iş test problemi üzerinde algoritmaların $f(x)$ min değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu.....	68

ÖZET

Anahtar kelimeler: Akış tipi çizelgeleme, genetik algoritmalar, çok kriterli karar verme, MultiMoora, sıra baskınlık teorisi, seçim stratejileri, eleme stratejileri

Bu çalışmada çözümünde sezgisel ve metasezgisel yöntemlerden yararlanılan, NP-Zor sınıfı çok amaçlı çizelgeleme problemlerinin bir türü olan, çok amaçlı akış tipi çizelgeleme problemi ele alınmıştır. Problemin çözümünde metasezgisel bir yöntem olan genetik algoritmalar tercih edilmiştir. Genetik algoritmaların çok kriterli değerlendirmedeki başarısını arttırmak amacıyla, çok kriterli karar verme yöntemlerinden MultiMoora, algoritmanın seçim, eleme ve hem seçim hem de eleme adımına birlikte entegre edilerek hibrit algoritmalar oluşturulmuştur. Önerilen hibrit algoritmalar farklı iş sayılarından oluşan maksimum tamamlanma zamanı, ortalama akış süresi, maksimum gecikme, ortalama gecikme ve geciken iş sayısı kriterlerinin birlikte ele alındığı çok amaçlı akış tipi çizelgeleme test problemleri üzerinde uygulanmış ve performansları çok amaçlı genetik algoritmalar ile karşılaştırılarak değerlendirilmiştir. Çalışma sonunda, genetik algoritmaların seçim adımına MultiMoora'nın entegre edilmesi ile oluşturulan hibrit algoritmalar ile başarılı sonuçlar elde edildiği görülmüştür.

SOLVING THE PROBLEM OF MULTI-OBJECTIVE SCHEDULING THROUGH GENETIC-MULTIMOORA HYBRID ALGORITHM

SUMMARY

Keywords: Flow-shop scheduling, genetic algorithm, multi criteria decision making, MultiMoora, theory of dominance, selection strategies, replacement strategies

In this paper, multi-objective flow-shop scheduling, which is a sort of NP-Hard multi-objective scheduling, was studied with the help of heuristic and metaheuristic method in its solution process. In this process, genetic algorithms are preferred as a metaheuristic method. In order to boost the success of the multi criteria analysis of genetic algorithms, hybrid algorithms are created by integrating MultiMoora –a multi criteria decision making method- into selection, replacement, and both replacement and selection steps. The suggested hybrid algorithms are applied on multi-objective flow-shop scheduling test problems formed of different job numbers, which are covered together with the criteria namely maximum makespan, average flowtime, maximum tardiness, average tardiness and number of late jobs; and their performances are evaluated comparing with multi-objective genetic algorithms. In the end of the study, it is seen that successful results are obtained by the use of hybrid algorithms composed by integrating MultiMoora into the selection step of genetic algorithms.

BÖLÜM 1. GİRİŞ

Çizelgeleme, imalat ve hizmet alanında önemli bir yeri olan, önemi günden güne artmaya devam eden bir karar sürecidir. Temelini kısıtlı olan kaynakların işlere atanması işlemi oluşturmaktadır. Özellikle imalat sektöründe rekabetin artması ile birlikte, çizelgeleme çalışmaları önem kazanmış ve çizelgeleme problemi olarak adlandırılan bu problem türüne çözüm sağlamak amacıyla çeşitli matematiksel ve sezgisel yöntemlerden yararlanılmıştır. Yıllar boyunca imalat, işlerin tek tezgahta üretimlerinin çizelgelenmesi üzerine çalışmalar yapılmış, ilerleyen yıllarda bu çalışmalar çok tezgahlı olarak ele alınmaya başlanmıştır. Bu durum çizelgeleme problemlerinin zorluk düzeyinin artmasına sebep olmaktadır. Ayrıca, başlangıçta tek ölçütün optimizasyonunun hedeflendiği tek amaçlı problemler olan çizelgeleme problemleri, daha hassas bir değerlendirme sağlayan ve birden fazla ölçütün birlikte optimizasyonunun hedeflendiği çok amaçlı problemler olarak ele alınmış ve bu durum problemin zorluk düzeyini daha da arttırmıştır. NP-Zor sınıfı olarak değerlendirilen problemler sınıfında yer alan bu çizelgeleme problemlerine, yaklaşık bir çözüm sağlamak amacıyla çeşitli sezgisel ve metasezgisel yöntemler geliştirilmiştir. Metasezgisel bir yöntem olan genetik algoritmaların çok amaçlı çizelgeleme problemlerinin çözümünde kullanıldığı birçok çalışma mevcuttur.

Çizelgeleme problemlerinin, yapılarındaki değişiklikler çerçevesinde farklı türleri bulunmaktadır. İmalat açısından değerlendirildiğinde, işlerin tezgahlarda aynı işlem sırasına sahip olduğu durumda akış tipi çizelgeleme olarak adlandırılan çizelgeleme problemi bu türlerden biridir. Akış tipi çizelgeleme üzerine yapılan çeşitli çalışmalar mevcuttur. Bu çalışmalar, başlangıçta maksimum tamamlanma zamanı C_{max} kriterinin optimizasyonunun hedeflendiği çalışmalar iken, bu kritere ortalama akış süresi \bar{F} , maksimum gecikme T_{max} gibi birçok kriterin optimizasyonu hedeflerinin de eklenmesi ile çok amaçlı akış tipi çizelgeleme problemleri olarak değerlendirilmiştir.

Çok amaçlı olarak ele alınan bu problemlerde birden fazla amacın optimizasyonunun sağlanmasına yönelik olarak problemin amaç fonksiyonu, problemde yer alan tüm kriterleri içeren şekilde oluşturulmaktadır. Böylece problemlere, tüm amaçları içeren tek bir amaç fonksiyonu ile çözüm aranmaktadır.

Bu çalışmada çok amaçlı çizelgeleme türlerinden biri olan çok amaçlı akış tipi çizelgeleme problemi ele alınmıştır. Maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve geciken iş sayısı NLJ olmak üzere 5 kriterin aynı anda optimizasyonu hedeflenmiştir. İşlerin 10 tezgahta işlem göreceği sıranın belirlenmesi hedeflenen problemde 15 iş, 25 iş, 50 iş ve 100 iş sayıları ile 4 farklı çok amaçlı akış tipi çizelgeleme problemine çözüm sağlanmaya çalışılmıştır. Çok iş, çok tezgah ve çok amaçlı olan NP-Zor sınıfı bu problemlere çözüm sağlamak için genetik algoritmalarından yararlanılmıştır. Genetik algoritmaların uygulamada çok amaçlı değerlendirmede performansını arttırmak amacıyla, genetik algoritmalara çok kriterli karar verme yöntemlerinden MultiMoora entegre edilmiştir. Bu entegrasyon işlemi 3 farklı şekilde tasarlanmıştır. Öncelikle MultiMoora, genetik algoritmaların seçim adımına farklı varyasyonlar ile entegre edilmiş ve 3 farklı hibrit algoritma oluşturulmuştur. Daha sonra MultiMoora, genetik algoritmaların eleme adımına entegre edilerek hibrit bir algoritma oluşturulmuştur. Son olarak da MutiMoora, genetik algoritmaların hem seçim hem de eleme adımına birlikte entegre edilmiş ve böylece bir hibrit algoritma daha elde edilmiştir. Genetik algoritmalara entegre edildikleri adımlar bazında 3 grup halinde incelenen 5 hibrit yapı, çok amaçlı genetik algoritma ile birlikte, 4 farklı iş sayısından oluşturulmuş çok amaçlı akış tipi çizelgeleme problemine uygulanmıştır. Önerilen algoritmaların performansları hem birbirleri ile hem de çok amaçlı genetik algoritmalar ile bu problemler üzerinde değerlendirilmiştir.

BÖLÜM 2. LİTERATÜR ÖZETİ

2.1. Akış Tipi Çizelgeleme Problemleri

Akış tipi çizelgeleme problemi üzerine yapılan çalışmaların başlangıcı, Johnson (1954) tarafından iki ve üç makineli akış tipi çizelgeleme probleminin ele alınmasına dayanmaktadır. 1990'lı yılların sonuna kadar problemler, genellikle tek amaçlı olarak değerlendirilmiştir. Bu problemlerde genellikle maksimum tamamlanma zamanı C_{max} kriterinin minimizasyonu üzerine çalışılmıştır (Rajendran ve Chaudhuri, 1990; Chen ve ark., 1995; Murata ve ark., 1996a; Das ve ark., 1995). Problemin birden fazla kriteri birlikte değerlendiren çok amaçlı problemler olarak ele alınmaya başlanması ile birlikte zorluk düzeyi artmış, bu nedenle çeşitli çözüm metotları uygulanmıştır.

Ishibuchi ve Murata (1996) tarafından maksimum tamamlanma zamanı, maksimum gecikme ve toplam akış zamanı kriterlerinin optimizasyonunda önerilen bir, çok amaçlı genetik lokal arama algoritması kullanılmıştır. Murata ve arkadaşları (1996b) tarafından amaçların ağırlıklarının değişken olduğu ve iyi bireylerin sonraki kuşağa aktarıldığı bir genetik algoritma önerilerek maksimum tamamlanma zamanı, toplam gecikme ve toplam akış zamanı kriterlerinin minimizasyonu sağlanmaya çalışılmıştır.

Genetik algoritma ile elde edilen çözümlere lokal arama uygulayan, her ebeveyn seçiminde farklı ağırlıklar kullanan bir algoritma önerilerek, maksimum tamamlanma zamanı ve maksimum gecikme kriterleri ile iki amaçlı, bu kriterlere toplam akış zamanı kriterinin de eklenmesi ile üç amaçlı problemlere uygulanmıştır. (Ishibuchi ve Murata, 1998).

Chang ve arkadaşları (2002) tarafından kademeli öncelikli ağırlıklandırma yaklaşımı önerilmiş, önerilen yaklaşımın etkinliği maksimum tamamlanma zamanı, toplam

gecikme ve toplam akış zamanı kriterlerinin optimize edildiği test problemi üzerinde değişken ağırlık yaklaşımı ile karşılaştırılarak değerlendirilmiştir.

Çok amaçlı genetik yerel arama algoritmasında değişiklik yapılarak Ishibuchi ve arkadaşları (2003) tarafından maksimum tamamlanma zamanı, maksimum gecikme kriterleri ile iki amaçlı ve toplam akış zamanı kriteri de bu kriterlere eklenerek oluşturulan üç amaçlı problemler üzerinde performansı test edilmiştir.

Maksimum tamamlanma zamanı ve gecikme kriterlerinin minimizasyonunda önerilen bir, mutasyon pareto genetik algoritması kullanılmıştır (Basseur ve ark., 2002). Pasupathy ve arkadaşları (2006) tarafından pareto sıralamaya dayalı yerel bir araştırma sağlayan ve çok amaçlı bir genetik algoritma olan PGA-ALS ile maksimum tamamlanma zamanı ve işlerin toplam akış süresinin optimizasyonu üzerine çalışılmıştır.

Li ve Wang (2007) tarafından maksimum tamamlanma zamanı ve maksimum gecikme kriterlerinin optimizasyonu, önerilen hibrit kuantumdan esinlenmiş bir genetik algoritma (HQGA) ile gerçekleştirilmiştir. Temiz ve Erol (2007) tarafından önerilen bulanık iş ve teslim zamanlı çok amaçlı genetik algoritma ile çizelge tamamlanma zamanı, maksimum tehir zamanı ve toplam akış zamanı kriterlerinin optimizasyonu sağlanmıştır.

Tavakkoli-Moghaddam ve arkadaşları (2007) tarafından önerilen hibrit çok amaçlı bir bağışıklık algoritması ile ağırlıklı ortalama tamamlanma süresi ve ağırlıklı ortalama gecikme kriterleri optimize edilmiştir. Geliştirilen çok objektif bir parçacık sürüsü (MOPS)'nün performansı, çok amaçlı genetik algoritma ile ağırlıklı ortalama tamamlanma süresi ve ağırlıklı ortalama gecikme kriterlerinin minimize edildiği problem üzerinde test edilmiştir (Rahimi-Vahed ve Mirghorbani, 2007).

Li ve arkadaşları (2008) tarafından önerilen bir hibrit çok objektif parçacık sürüsü optimizasyonu (MOPSO) algoritması, maksimum tamamlanma zamanı, ortalama tamamlanma süresi ve maksimum gecikme kriterlerinin optimizasyonunun

hedeflendiği problem üzerinde çeşitli yöntemlerle karşılaştırılarak test edilmiştir. Maksimum tamamlanma zamanı ve maksimum gecikme kriterleri, Chang ve arkadaşları (2008) tarafından SGA ve NSGAI ile gömülü yapay kromozomlar kullanılarak geliştirilen bir genetik algoritma ile minimize edilmiştir.

Ağırlıklı ortalama erken bitme ve ağırlıklı ortalama gecikme kriterlerinin optimizasyonunda, önerilen melez çok amaçlı karıştırılmış sıçrayan kurbağa algoritması kullanılmış, önerilen algoritmanın etkinliği çok amaçlı genetik algoritmalarla karşılaştırılarak gösterilmiştir (Rahimi-Vahed ve ark., 2009). Diferansiyel evrime dayalı hibrit bir algoritma önerilmiş ve önerilen algoritmanın etkinliği, maksimum tamamlanma zamanı ve maksimum gecikme kriterlerinin optimizasyonunun hedeflendiği problem üzerinde test edilmiştir (Pan ve ark., 2009).

Yeniden girişli hibrit akış tipi çizelgeleme problemine çevrim süresi ve darboğaz kullanım oranı kriterleri açısından, önerilen Lorenz dominantlık ilişkisini kullanan L-NSGA adlı yeni bir, çok amaçlı genetik algoritma ile çözüm sağlanmıştır (Dugardin ve ark., 2010).

Maksimum tamamlanma zamanı ve maksimum gecikme kriterlerinin değerlendirildiği bağlantısız paralel makinelerle hibrit akış tipi çizelgeleme problemlerinin çözümünde, geliştirilen hibrit çok amaçlı paralel bir genetik algoritma kullanılmıştır. (Rashidi ve ark., 2010).

Keskin (2010) tarafından toplam akış zamanı ve maksimum tamamlanma zamanı kriterlerinin optimizasyonunun hedeflendiği beklemez akış tipi çizelgeleme problemine, geliştirilen çok amaçlı melez genetik algoritma ile çözüm sağlanmıştır.

Temiz (2010) tarafından üç aşamalı tavlama benzetimi algoritması geliştirilmiş, tamamlanma zamanı, toplam akış zamanı ve en büyük tehir zamanı kriterlerinin yer aldığı permütasyon akış tipi çizelgeleme problemine uygulanmıştır.

Mokhtari ve arkadaşları (2011) tarafından Hibrit Ayırık Diferansiyel Evrim Algoritması ve Değişken Mahalle Araması'nın kombinasyonu ile önerilen model kullanılarak, maksimum tamamlanma zamanı ve toplam kaynak maliyeti kriterleri minimize edilmiştir.

Zandieh ve Karimi (2011) tarafından çok popülasyonlu bir genetik algoritma önerilmiş, önerilen algoritmanın başarısı MOGA ve NSGA-II algoritmaları ile toplam ağırlıklı gecikme ve tamamlanma süresi kriterleri açısından test edilmiştir.

Gecikmelerin ağırlıklı kareleri toplamı, maksimum tamamlanma zamanı, erken bitirmelerin ağırlıklı kareleri toplamı ve geciken iş sayısı kriterlerinin optimizasyonu, Majazi ve arkadaşları (2011) tarafından sunulan, hibrit bir genetik algoritma ile sağlanmıştır.

Pour ve arkadaşları (2013) tarafından önerilen yeni bir genetik algoritma maksimum tamamlanma zamanı, toplam bekleme süresi ve toplam gecikme kriterlerinin optimizasyonunda kullanılmıştır.

Marichelvam ve arkadaşları (2014) tarafından maksimum tamamlanma zamanının ağırlıklı toplamı ve ortalama akış süresi kriterlerinin minimizasyonu ateşböceği algoritması ile sağlanmıştır.

Hosseini (2017) tarafından önerilen Pareto optimal sınırdaki yerel arama için, genetik algoritma temelli çok amaçlı bir algoritma kullanılarak maksimum tamamlanma zamanı, toplam erken bitirme ve gecikme kriterlerinin eş zamanlı minimizasyonu sağlanmıştır.

Yapılan çalışmalar incelendiğinde çok amaçlı akış tipi çizelgeleme problemlerinin çözümünde genetik algoritmalarla sıklıkla yararlanıldığı görülmektedir.

2.2. Genetik Algoritmalar ve Çok Kriterli Karar Verme Yöntemleri

Akış tipi çizelgeleme problemlerinin çok amaçlı olarak değerlendirildiği durumlarda, probleme çözüm elde etmek amacıyla genetik algoritmaların çok kriterli karar verme metotları ile birlikte kullanıldığı çeşitli çalışmalar mevcuttur.

Braglia ve Grassi (2009) tarafından maksimum tamamlanma zamanı ve maksimum gecikme kriterlerinin optimizasyonunda Topsis yöntemi ile NEH (Nawaz–Enscore–Ham) sezgiseli birleştirilerek uygulanmış, uygulanan modelin performansı çok amaçlı genetik yerel arama algoritması ile karşılaştırılarak test edilmiştir.

Lin ve arkadaşları (2011) tarafından yeniden girişli akış tipi çizelgeleme probleminin çözümünde ortalama ve maksimum tamamlanma zamanı sapması, zamanında sipariş oranı kriterleri AHP ve genetik algoritma birlikte kullanılarak optimize edilmiştir. Yine aynı yazarlar tarafından, AHP ve genetik algoritma ile ortalama ve maksimum tamamlanma zamanı sapması, zamanında sipariş oranı kriterlerinin optimizasyonunun hedeflendiği yeniden girişli akış tipi çizelgeleme problemine çözüm sağlanmıştır (Lin ve ark., 2012).

BÖLÜM 3. ÇİZELGELEME

Çizelgeleme, var olan kısıtlı kaynakların işlere optimum şekilde atanması olarak ifade edilebilmektedir. Çizelgeleme konusunda yapılan çalışmaların başlangıcı 1950'lere dayanmaktadır (Kamışlı Öztürk ve Sağır Özdemir, 2005). Endüstrinin gelişmesi ve işletmeler arası rekabetin artması, müşteri memnuniyetinin minimum kaynak kullanımı ile sağlanmasının önem kazanmasına ve bununla birlikte özellikle üretim alanında çizelgeleme çalışmalarının artışına yol açmıştır. Çizelgeleme çalışmaları işlerde meydana gelen gecikmeler, üretim maliyetleri, hazırlık zamanları, makine boşa kalma zamanı gibi çeşitli faktörlerin bir veya birden fazlasının minimum düzeye indirilmesi hedefi doğrultusunda yapılmaktadır (Eren ve Güner, 2005). Üretim alanında n tane işin tezgahta hangi sıra ile işleneceğine yönelik literatürde çeşitli çalışmalar mevcuttur. Yıllar boyunca imalat alanında, yapılan işlerin tek tezgahta işlenme sıralarının belirlenmesi üzerine çizelgeleme çalışmaları yapılmıştır. İşletmeler arası rekabetin artması, çalışmalarda birden fazla kriterin birlikte değerlendirilmesi ile bu kriterlerin birlikte optimizasyonunun hedeflendiği çizelgeleme çalışmalarını ortaya çıkarmıştır. Örneğin, iki kriterin birlikte minimizasyonunun hedeflendiği bir minimizasyon probleminde, kriterlerin tek başına minimum seviyelere ulaştırılmasındansa ortak sonucun minimizasyonu hedeflenmektedir. Birden fazla kriteri birlikte değerlendiren bu çalışmalarda çizelgeleme problemi, çok amaçlı çizelgeleme problemi olarak adlandırılmaktadır.

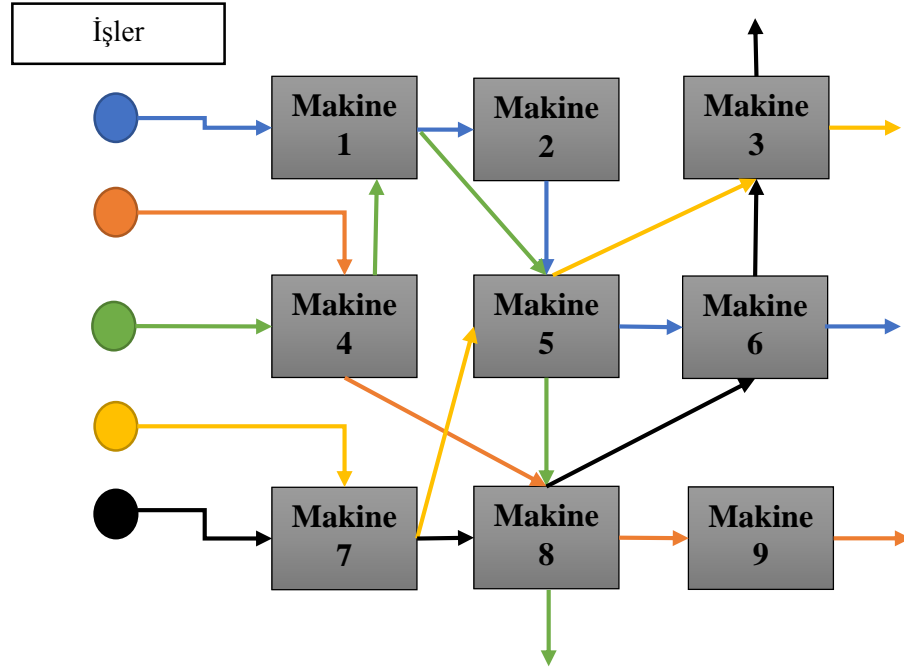
Literatürde birden fazla kriterin birlikte değerlendirildiği çizelgeleme problemlerinin çeşitli alanlarda uygulamaları yer almaktadır. Sağlık ve hizmet sektörleri, üretim alanı ve araç rotalama problemleri gibi çeşitli alanlarda çok amaçlı çizelgeleme problemlerinin çözümüne yönelik çalışmalar mevcuttur (Kamışlı Öztürk ve Sağır Özdemir, 2005).

Çizelgeleme problemlerinin yapısındaki bazı farklılıklardan dolayı birçok çeşidi mevcuttur. Parametrelerinin belirli olup olmaması çizelgeleme problemlerini kategorize etmeyi sağlayan bir ölçüttür. Belirli ise deterministik, belirsiz ise stokastik olarak gruplandırılmaktadır. Çizelgeleme problemlerinde işlerin başlangıçta belirli olması veya değişken olması da problemlerin gruplandırılmasında değerlendirme sağlamaktadır. Belirli ise statik çizelgeleme, belirsiz ise dinamik çizelgeleme olarak adlandırılmaktadır (Gözen, 2007). İşlerin makinelerde işlendiği sıranın aynı olması veya farklılık gösterebilmesi durumuna göre ise çizelgeleme iki grupta incelenmektedir. İşler aynı sırada işleniyorsa akış tipi çizelgeleme, farklı sıralarda işlem görebiliyorsa bu durumda çalışılan çizelgeleme, bir atölye tipi çizelgelemedir. Atölye tipi ve akış tipi çizelgeleme aşağıda detaylandırılmıştır.

3.1. Atölye Tipi Çizelgeleme

İmalatta çizelgeleme çalışmalarında *is* sayıda işin, *ms* sayıda tezgahta farklı sıralarda işlem gördüğü durumda yapılan çizelgeleme, atölye tipi çizelgeleme olarak adlandırılmaktadır. Çalışmalarda amaç, işlere uygulanacak tüm işlemlerin başladığı en uygun zamanın belirlenmesidir. Atölye tipi çizelgeleme çalışmalarında işlerin başlangıçta işlenmek üzere hazır bulunduğu varsayımı kabul edilmektedir. İş ve tezgah sayıları, işlem süreleri başlangıçta belirlidir. İşler tezgahların her birinde yalnız bir kez işlem görecektir şekilde operasyonlara sahiptir (Şevkli, 2005).

Atölye tipi çizelgelemede her işin kendine ait diğer işlerden bağımsız bir rotaya ve işlem adımlarına sahip olması, problemin zorluk düzeyini arttırmaktadır (Dilaver, 2015). Çalışmalarda iş ve tezgah sayısının da artması ile problem, çözümü oldukça zor ve NP-Zor sınıfı bir problem haline gelmektedir. Bu nedenle çözümünde, NP-Zor sınıfı problemlere yaklaşık çözümler sağlamak amacıyla geliştirilen çeşitli sezgisel ve metasezgisel yöntemler sıklıkla tercih edilmektedir. Problemin çözümünün ifade edilmesinde genellikle Gantt şemalarından yararlanılmaktadır (Aydemir, 2009). Atölye tipi çizelgelemenin genel yapısı Şekil 3.1.'de gösterilmiştir.



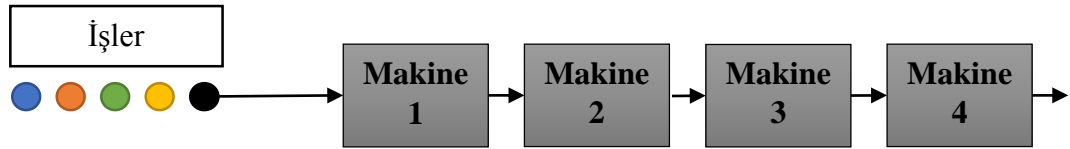
Şekil 3.1. Atölye tipi çizelgeleme

3.2. Akış Tipi Çizelgeleme

İşlerin tezgahlarda aynı sıra ile işlendiği çizelgeleme türü olan akış tipi çizelgeleme, üretim alanında yaygın olarak kullanılan bir çizelgeleme türüdür. is sayıda işin $j_1, j_2, j_3, \dots, j_{is}$, ms sayıda tezgahta hangi sırada işleneceğinin önceden belirli olduğu ve işlerin birbirlerine göre işlem görme önceliklerinin belirlendiği çalışmalardır. Literatürde akış tipi çizelgeleme problemi olarak yer alan bu problem türüne çözüm elde edilirken çeşitli varsayımlar kabul edilmektedir. Tezgahlarda aynı anda yalnızca bir işin işlem görebilmesi, işlere ait hazırlık zamanlarının işlem süreleri içerisine dahil kabul edilmesi, tezgahların kesintisiz olarak çalışmaya devam etmesi, işlerin tezgahlardaki işlem sürelerinin önceden biliniyor olması ve tüm işlerin işlenmek için başlangıç anında hazır bulunması bu varsayımlardandır (Yağmahan ve Yenisey, 2006).

Akış tipi çizelgeleme problemleri önceleri tek tezgahlı problemler olarak değerlendirilmiş, daha sonra çok tezgahlı problemlere çözüm getirilmeye çalışılmıştır.

Ayrıca akış tipi çizelgeleme problemlerinde her bir işin i , her bir tezgahdaki j önceden belirli olan işlem süresi t_{ij} göz önünde bulundurularak, işin önceden belirli olan teslim tarihinden d_i önce tamamlanması gerekmektedir. Tüm bu durumlar değerlendirildiğinde akış tipi çizelgeleme problemlerinde işlem göreceğ iş sayısının ve tezgah sayısının fazla olması problemi zorlaştırmaktadır. Bu tür problemler NP-Zor sınıfı problemler olarak adlandırılmaktadır (Ponnambalam ve ark., 2004). NP-Zor sınıfı problemlere çözüm getirmek amacıyla, probleme yaklaşık bir çözüm elde edilmesini sağlayan metotların kullanımı mevcuttur. Genellikle sezgisel ve metasezgisel yöntemler bu problem türünün çözümünde tercih edilmektedir. Akış tipi çizelgeleme problemlerinin genel yapısı Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Akış tipi çizelgeleme

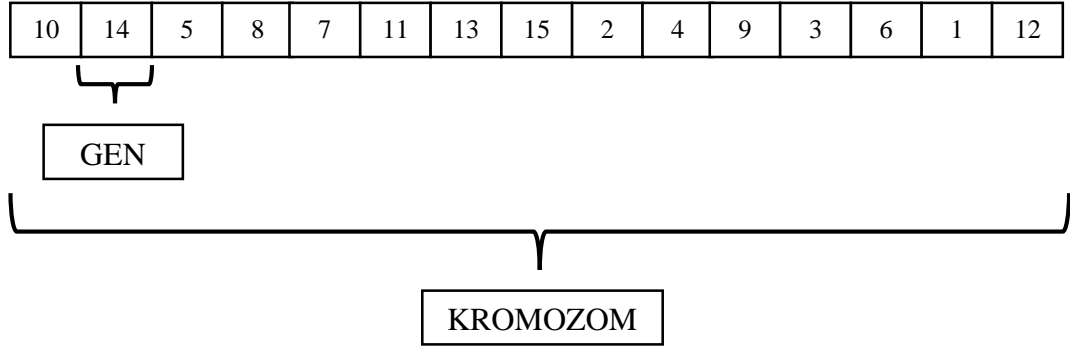
Akış tipi çizelgeleme problemlerinin ilk çalışmalarında problemin tek amacı optimizasyonu hedeflenmiş, büyük bir çoğunluğunda maksimum tamamlanma zamanının C_{max} minimizasyonu sağlanmaya çalışılmıştır. Birçok amacın birlikte değerlendirildiği çalışmaların yapılması ile birlikte C_{max} kriterinin yanında ortalama akış süresi \bar{F} , maksimum gecikme T_{max} gibi kriterler de değerlendirmede kullanılmış ve problem, çok amaçlı akış tipi çizelgeleme problemi olarak ele alınmıştır. Çok amaçlı olarak uygulanan bu çalışmalarda optimizasyonu hedeflenen fonksiyon, değerlendirilecek tüm kriterleri içerecek şekilde tasarlanmaktadır.

BÖLÜM 4. METOTLAR

Çalışmada genetik algoritmaların seçim, eleme ve hem seçim hem de eleme adımlarına birlikte MultiMoora metodu entegre edilerek 3 grup olarak incelenen, 5 farklı hibrit algoritma oluşturulmuştur. Çalışmanın bu bölümünde, hibrit olarak kullanılan, genetik algoritmalar ve MultiMoora metodu detaylandırılmıştır.

4.1. Genetik Algoritmalar

Charles Darwin'in en iyi olanın varlığını sürdürmesi ilkesine dayanan genetik algoritmalar, ilk kez 1975 yılında Holland tarafından önerilmiştir (Poon ve Carter, 1995; Gonçalves ve ark., 2005). Goldberg ile bilinen bir metasezgisel yöntem halini almıştır (Pour ve ark., 2013). Genetik algoritmalarda öncelikle, olası çözümler gen adı verilen birimlerden meydana gelen kromozomlar ile kodlanmaktadır. Kromozomlarda, genetik algoritmaların uygulandığı problemin yapısı gereği, kullanılan kodlama türü farklılık göstermektedir. İkili kodlama ve permütasyon kodlama genetik algoritmalarda kullanımı yaygın olan kodlama türleridir (Çolak, 2010). Çalışmada 15 işten oluşan çok amaçlı akış tipi çizelgeleme probleminde, işlerin tezgahlarda işlenme sıralarını ifade eden kromozoma ait kodlama örneği Şekil 4.1.'de gösterilmiştir.



Şekil 4.1. 15 işten oluşan çok amaçlı akış tipi çizelgeleme problemi için kromozom örneği

Algoritma uygulanırken, olası çözümlerden oluşan kromozomlar rassal olarak, istenen popülasyon büyüklüğü kadar türetilir ve başlangıç popülasyonu oluşturulur. Uygun bir seçim stratejisi kullanılarak, oluşturulan başlangıç popülasyonundan çaprazlama adımına gönderilecek bireyler seçilir ve seçilen çaprazlama stratejisi doğrultusunda çaprazlama işlemi uygulanır. Çaprazlama işleminin ardından uygun bir mutasyon stratejisi seçilerek mutasyon işlemi uygulanır. Çaprazlama ve mutasyon işlemlerinin uygulanması ile yeni jenerasyon bireyler elde edilir. Bir eleme stratejisi belirlenerek popülasyondaki hangi eski bireylerin yerini yeni bireylerin alacağı belirlenir ve yeni bireyler popülasyona aktarılır. Bu işlemlerin tekrar etmesi ile her jenerasyonda daha iyi bireyler elde edilmeye ve çözüm kalitesi arttırılmaya çalışılır. Belirlenen durdurma kriterinin sağlanması ile işlemler sonlandırılır ve olası en iyi çözüm elde edilmiş olur.

Genetik algoritmaların, tek amacın optimizasyonunun hedeflendiği problemlere uygulandığı çeşitli çalışmalar mevcuttur. Gelişen endüstri ile birden fazla amacın birlikte değerlendirilmesi ihtiyacı meydana gelmiş ve genetik algoritmalar, çok amaçlı problemlerde uygulanmaya başlanmıştır. Çok amaçlı genetik algoritmalar 1984 yılında Schaffer tarafından Vektör Değerli Genetik Algoritma (VEGA) olarak önerilmiştir (Horn ve ark., 1994). Genetik algoritmalar uygulanırken kullanılan uygunluk fonksiyonu, çok amaçlı problemlerde yapılan uygulamalarda, optimizasyonu hedeflenen tüm kriterlerin birlikte yer aldığı çok amaçlı bir fonksiyon

olarak oluşturulmaktadır (Veeraiah ve ark., 2017). Çalışmada yararlanılan 5 kriterin birlikte optimizasyonunun hedeflendiği fonksiyon, denklem (4.1)'de yer almaktadır.

$\alpha, \beta, \gamma, \delta, \theta$ Kriter ağırlıkları olmak üzere,

$$f(x) = \text{Min}[\alpha C_{max} + \beta \bar{F} + \gamma T_{max} + \delta \bar{T} + \theta NLJ] \quad (4.1)$$

$$\alpha \geq 0, \beta \geq 0, \gamma \geq 0, \delta \geq 0, \theta \geq 0$$

$$\alpha + \beta + \gamma + \delta + \theta = 1$$

C_{max} = Maksimum tamamlanma zamanı

\bar{F} = Ortalama akış süresi

T_{max} = Maksimum gecikme

\bar{T} = Ortalama gecikme

NLJ = Geciken iş sayısı

Genetik algoritmaların çalışma performansı popülasyon büyüklüğünden ve tercih edilen seçim, çaprazlama, mutasyon, eleme stratejilerinden etkilenmektedir. Genetik algoritmaların temel adımları olan seçim, çaprazlama, mutasyon ve eleme adımları aşağıda detaylandırılmıştır.

4.1.1. Seçim

Genetik algoritma uygulamalarında çaprazlama ve mutasyon işlemlerinin uygulanacağı bireyler, popülasyonda yer alan bireyler arasından seçilmektedir. Seçilecek bireyin belirlenmesi konusunda geliştirilmiş çeşitli seçim stratejileri mevcuttur. Seçim stratejisi tercihi, algoritma performansını büyük ölçüde etkilediğinden önemli bir karardır. Literatürde kullanımı yaygın olan Rulet Seçimi, Lineer Rank Seçimi ve Turnuva Seçimi seçim stratejileri açıklanmıştır.

4.1.1.1. Rulet tekerleği seçimi

Holland tarafından 1975 yılında önerilen bir seçim stratejisi olan Rulet Tekerleği Seçimi, genetik algoritmaların önerilen ilk seçim stratejisidir (Abd Rahman ve ark., 2016). Rulet Tekerleği Seçimi'nin ilk adımını, başlangıç popülasyonunda yer alan

bireylerin uygunluk fonksiyonu ile uygunluk değerlerinin hesaplanması oluşturmaktadır. Popülasyonda yer alan bireylere, hesaplanan uygunluk değerleri oranınca $[0,100]$ aralığında oluşturulan rulet çemberi üzerinde alan sağlanır. Uygunluk fonksiyonu değeri yüksek olan birey, rulet çemberi üzerinde daha fazla alana sahip olur ve seçilme şansı artar. Rassal sayı türetilerek, bu rassal sayının rulet çemberi üzerinde karşılık geldiği alandaki kromozom, çaprazlama işlemine gönderilmek üzere seçilir. İşlemlerin çaprazlama oranınca tekrarlanması ile çaprazlama işlemi uygulanacak çapraz çiftleri belirlenmiş olur.

4.1.1.2. Lineer rank seçimi

Rank Seçimi ilk kez Baker tarafından orantılı seçimin olumsuz yönlerini ortadan kaldırmak amacıyla önerilmiş bir seçim stratejisidir (Blickle ve Thiele, 1995). Seçim stratejisinin ilk adımını Rulet Tekerleği Seçimi'nde olduğu gibi başlangıç popülasyonunda yer alan kromozomların uygunluk fonksiyonu ile uygunluk değerlerinin hesaplanması oluşturmaktadır. Popülasyonda yer alan bireyler en kötü uygunluk değerine sahip bireyden en iyi uygunluk değerine sahip bireye doğru sıralanır. Uygunluk değeri en kötü olan bireye sıra numarası olarak 1 verilecek şekilde en iyi bireye doğru 1 arttırılarak popülasyondaki birey sayısına kadar tüm kromozomlara sıra numarası verilir. Verilen sıra numaraları toplanır ve sırasıyla tüm kromozomların sıra numarası bu toplama oranlanır. Elde edilen oranlar düzeyinde bireyler için seçilme aralıkları belirlenir. Rassal sayı türetilir ve türetilen sayının karşılık geldiği alanda yer alan kromozom çaprazlama işlemi uygulanacak çapraz çiftinin ilk bireyi olarak seçilir. İşlemlerin çaprazlama oranınca tekrarlanması ile çaprazlama işlemi uygulanacak bireyler tamamlanmış olur.

4.1.1.3. Turnuva seçimi

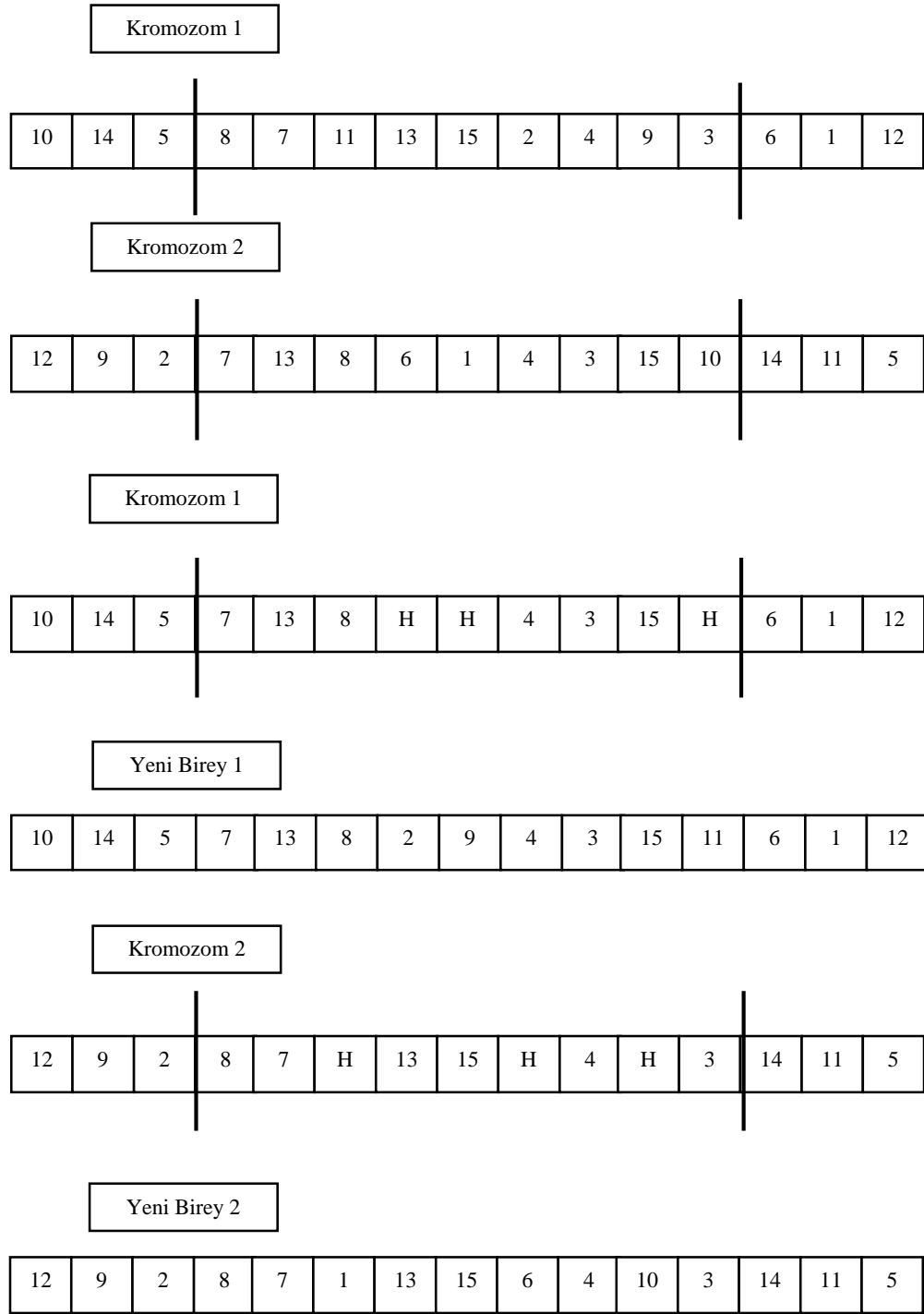
Turnuva Seçimi seçim stratejisi, Brindle tarafından 1980'lerin başında yapılan çalışma ile ortaya çıkmıştır (Xie ve ark., 2008). Turnuva Seçimi uygulanırken, başlangıç popülasyonunda yer alan bireylerden, belirlenen bir turnuva genişliği sayısı kadar bireyin seçilmesi ilk adımı oluşturmaktadır. Seçilen turnuva genişliği kadar bireyin

uygunluk fonksiyonu ile uygunluk deęerleri hesaplanır. Hesaplanan uygunluk deęerleri karřılařtırıldığında, en iyi uygunluk deęerinin elde edildięi birey aprazlama iřlemine gnderilmek zere seilir. İřlemler aprazlama oranınca tekrarlanır ve ihtiya duyulan tm apraz iftleri oluřturulur.

4.1.2. aprazlama

aprazlama, genetik algoritmaların poplasyonda yer alan bireylerden yeni jenerasyon bireyler elde etmek amacıyla uygulanan bir adımdır. aprazlama iřlemi ile poplasyon kalitesinin nesilden nesile arttırılması hedeflenmektedir. Literatrde eřitli problemlerde genetik algoritmaların uygulanması sırasında kullanılan, birok aprazlama stratejisi bulunmaktadır. Pozisyona dayalı aprazlama, sıraya dayalı aprazlama, sıralı aprazlama ve doęrusal sıralı aprazlama yaygın olarak kullanılan aprazlama stratejilerinden bazılarıdır (Gerřil ve Palamutuoęlu, 2013). Genetik algoritmaların uygulanması sırasında tercih edilen aprazlama yntemi, algoritma performansı zerinde olduka etkilidir.

alıřmada aprazlama iřlemi, doęrusal sıralı aprazlamanın farklı bir řekilde uygulanması ile gerekleřtirilmiřtir. aprazlama iřlemi uygulanırken ncelikle, seilen kromozom ifti zerinde hangi genler arasında deęiřiklik yapılacağı belirlenmektedir. Belirlenen bu aralık, kromozomlar zerinde iřaretlenmektedir. Deęiřtirilecek aralıkta yer alan genlerin, kromozomlarda karřılıklı olarak yer deęiřirmesi saęlanmaktadır. Kromozomlar zerinde aynı genin bulunması halinde, kromozoma yeni aktarılan blmde, akıřan genlerin zerine H harfi yerleřtirilmektedir. Kromozom zerindeki tm genlerin incelenmesi ile H harfi yazılması gereken genler tamamlanmaktadır. Kromozomda yer almayan gen numaraları kkten byęe doęru, kromozomun en solunda yer alan H'den bařlanarak H harflerinin yerine yerleřtirilmektedir. Seilen iki kromozom iin de aynı iřlemin uygulanması sonucu, yeni jenerasyon bireyler elde edilmektedir. řekil 4.2.'de 15 iřin izelgelendięi rnek bir kromozom zerinde, doęrusal sıralı aprazlamanın alıřmada kullanılan farklı uygulaması gsterilmektedir.

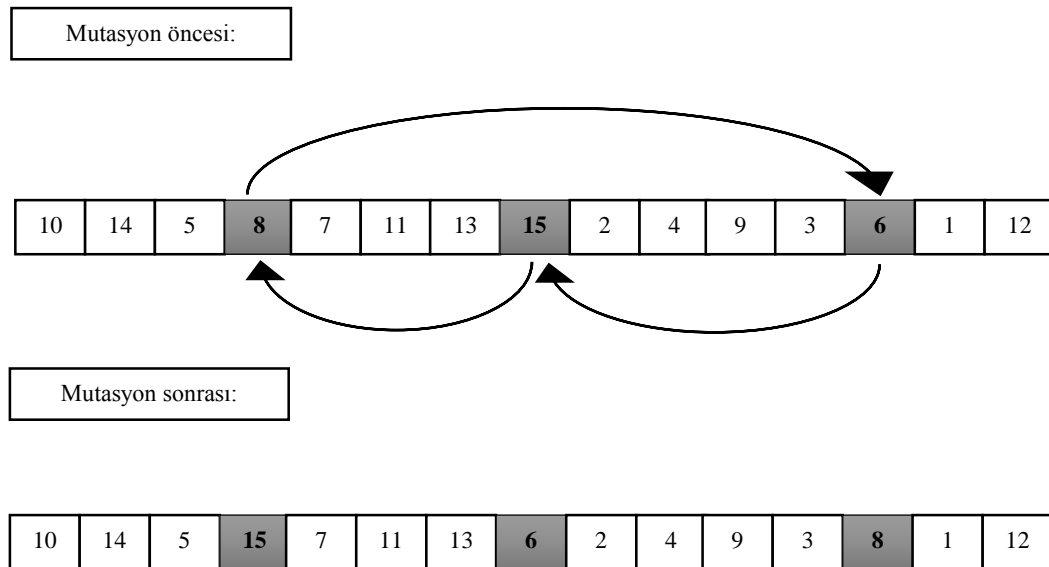


Şekil 4.2. 15 için çizelgelendiği örnek bir kromozom üzerinde doğrusal sıralı çaprazlamanın farklı bir uygulaması

4.1.3. Mutasyon

Mutasyon, kromozomların rassal olarak belirlenen gen veya genleri üzerinde değişiklik yapılması işlemidir. Genetik algoritmalarda, genetik çeşitliliğin korunmasının sağlanması ve problem üzerinde çözüm elde etmenin kolaylaşması amacıyla uygulanmaktadır. Kromozomlara mutasyon işlemi uygulanması ile yeni bireyler elde edilir ve nesilden nesile daha iyi bireylerin elde edilmesi sağlanmaya çalışılır. Genetik algoritma uygulamalarında, tercih edilen mutasyon stratejisi algoritma performansını etkilemektedir. Literatürde geliştirilen çeşitli mutasyon stratejileri mevcuttur. Ters mutasyon, komşu iki geni değiştirme, keyfi iki geni değiştirme ve keyfi üç geni değiştirme kullanılan bazı mutasyon stratejilerinden.

Çalışmada, literatürde yer alan mutasyon stratejilerinden biri olan keyfi üç geni değiştirme mutasyonu uygulanmıştır. Bu mutasyon stratejisi, kromozom üzerinde rassal olarak belirlenen üç genin birbirleri ile yer değiştirilmesi şeklinde uygulanmaktadır. 15 işin çizelgelendiği çok amaçlı akış tipi çizelgeleme problemine ait örnek bir kromozom üzerinde, keyfi üç geni değiştirme mutasyonunun uygulanması Şekil 4.3.'te gösterilmiştir.



Şekil 4.3. 15 işin çizelgelendiği örnek bir kromozoma keyfi üç geni değiştirme mutasyonu uygulanması

4.1.4. Eleme

Genetik algoritmelerde çaprazlama ve mutasyon işlemleri sonucu üretilen yeni jenerasyon bireyler popülasyona aktarılır. Bu işlem sırasında popülasyon büyüklüğünü sabit tutmak amacıyla, popülasyona aktarılabilecek yeni birey sayısı kadar eski birey popülasyondan elenir. Popülasyondan elenen bireylerin yerine yeni jenerasyon bireyler popülasyona aktarılır. Böylece popülasyon büyüklüğü korunarak yeni bireylerin popülasyonda yer alması sağlanır. Eski bireylerin elenmesi sırasında hangi eleme stratejisinin tercih edildiği genetik algoritmaların çalışma performansını etkilemektedir. Bu nedenle geliştirilmiş çeşitli eleme stratejileri mevcuttur. Bu stratejilerden En Kötüyü Eleme, Rastgele Eleme ve En Eski Olanı Eleme stratejileri yaygın olarak kullanılanlardandır (Wu ve ark., 2014).

4.1.4.1. En kötüyü eleme

Genetik algoritma uygulamalarında sıklıkla tercih edilen eleme stratejilerinden en kötüyü eleme stratejisi, üretilen yeni jenerasyon bireylerin popülasyonda mevcut olan uygunluk değeri en kötü bireyler ile yer değiştirilmesine dayanmaktadır. Böylece popülasyondan kötü olan bir birey uzaklaştırılmış ve popülasyona yeni jenerasyon bir birey kazandırılmış olur. İşlemler sonucu başlangıçta belirlenen popülasyon büyüklüğü korunur.

4.1.4.2. Rastgele eleme

Rastgele eleme stratejisi, çaprazlama ve mutasyon işlemlerinin ardından oluşturulan yeni jenerasyon bireylerin, popülasyonda önceden yer alan bireylerden rassal olarak seçilenlerinin yerine popülasyona aktarılması şeklinde uygulanmaktadır. Popülasyon büyüklüğünün az olduğu problemlerde uygulandığında diğer problemlere göre daha iyi sonuçlar elde edilebilmektedir.

4.1.4.3. En eski olanı eleme

En eski olanı eleme stratejisi, genetik algoritma operatörleri kullanılarak oluşturulan yeni jenerasyon bireylerin popülasyona aktarılması sırasında, popülasyonda yer alan bireylerden en eski olanın popülasyondan elenmesi işlemine dayanmaktadır. Popülasyonun en eski bireyinin elenmesi ve yerini yeni bireyin alması ile popülasyonda her jenerasyonda iyileşme sağlanması hedeflenmektedir.

4.2. MultiMoora Metodu

MultiMoora (Moora plus the full multiplicative form) metodunun temeli Brauers ve Zavadskas (2006) tarafından Moora (the multi-objective optimization on the basis of ratio analysis) metodunun geliştirilmesine dayanmaktadır. Moora metoduna daha sonra Tam Çarpım Formu metodu eklenerek MultiMoora metodu oluşturulmuştur (Brauers ve Zavadskas, 2010). MultiMoora metodu, Moora metodu içerisinde yer alan Oran Metodu, Referans Noktası Yaklaşımı metotlarının ve bu metotlara daha sonra eklenen Tam Çarpım Formu metodunun sonucunun kıyaslanarak, Sıra Baskınlık Teorisi ile değerlendirilmesi şeklinde uygulanmaktadır.

4.2.1. Moora-oran metodu

Moora metodunun ilk adımını oluşturan Oran Metodu uygulanırken öncelikle, değerlendirilecek alternatiflerin değerlendirme işleminde yararlanılacak kriterler bazında performans değerlerinden oluşan bir başlangıç matrisi oluşturulmaktadır. Bu matriste etkisi olumlu yönde olan kriterler maksimum, etkisi olumsuz olan kriterler ise minimum kriter olarak matris üzerinde gösterilir (4.2).

$$X = \begin{matrix} & \begin{matrix} C_1 & C_2 & \dots & C_n \end{matrix} \\ \begin{matrix} A_1 \\ A_2 \\ \dots \\ A_m \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \end{matrix} \quad (4.2)$$

1.Adım: Alternatifler, kriterler ve alternatiflerin kriterler bazında performans değerlerinden oluşan başlangıç matrisi oluşturulur. Maksimum ve minimum kriterler belirlenir.

2.Adım: Oluşturulan başlangıç matrisine, normalizasyon formülü (4.3) kullanılarak normalizasyon işlemi uygulanır (Datta ve ark., 2013). Normalize edilmiş değerlerin x_{ij}^* yer aldığı yeni bir matris oluşturulur.

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\sum_{j=1}^m x_{ij}^2}} \quad (4.3)$$

$i = 1, 2, \dots, n$; n kriterlerin sayısı

$j = 1, 2, \dots, m$; m alternatiflerin sayısı

$x_{ij} = j$. alternatifin i . kriter açısından performans ölçüm değeri

3.Adım: Oluşturulan normalize edilmiş matriste yer alan tüm değerler, bulundukları sütunlardaki kriterlere ait kriter ağırlığı w_i ile çarpılır (4.4). Böylece ağırlıklandırılmış değerlerden v_{ij} oluşan yeni bir matris elde edilir.

$$v_{ij} = w_i * x_{ij}^* \quad (4.4)$$

$w_i = i$. kriter ağırlığı

4.Adım: Normalizasyon ve ağırlıklandırma işlemleri sonucu elde edilen yeni matriste, her bir alternatif için maksimum kriterler toplamından minimum kriterler toplamı çıkarılarak her bir alternatif için Oran Metodu puanı y_j^* elde edilir (4.5).

$$y_j^* = \sum_{i=1}^g v_{ij} - \sum_{i=g+1}^n v_{ij} \quad (4.5)$$

$i = 1, \dots, g$ maksimize edilecek hedefler

$i = g + 1, \dots, n$ minimize edilecek hedefler

$g =$ maksimize edilecek hedeflerin sayısı

$n - g =$ minimize edilecek hedeflerin sayısı

$y_j^* = j$ alternatifinin tüm kriterlere göre normalize edilmiş değeri

5.Adım: Oran metodu puanları büyükten küçüğe doğru sıralanır. Elde edilen sıralama Oran Metodu sonucunda alternatiflere ait sıralamayı ifade eder.

4.2.2. Referans noktası yaklaşımı

Referans Noktası Yaklaşımı, Moora metodunun ikinci adımını oluşturmaktadır. Oran Metodu'nda olduğu gibi başlangıç matrisine normalizasyon formülü (4.3) ile normalizasyon, ardından ağırlıklandırma formülü (4.4) ile ağırlıklandırma işleminin uygulanması sonucu elde edilen matristen yararlanılır. Bu matrise Referans Noktası Yaklaşımı formülleri uygulanır ve metoda ait sonuçlar elde edilir. Referans Noktası Yaklaşımı işlem basamakları aşağıda sıralanmaktadır:

1.Adım: Normalizasyon ve ağırlıklandırma işlemleri sonucu elde edilen matriste maksimum kriter sütunlarının maksimum değeri, minimum kriter sütunlarının ise minimum değeri referans değeri r_i olarak belirlenir.

2.Adım: Matristeki tüm değerlerin, bulundukları sütuna ait referans değeri ile farkının mutlak değeri alınarak (4.6), elde edilen bu değerlerden yeni bir matris oluşturulur.

$$|w_i r_i - v_{ij}| \quad (4.6)$$

3.Adım: Oluşturulan yeni matriste her bir alternatif için satır maksimum değeri tespit edilir. Bu değerlerin küçükten büyüğe doğru sıralanması (4.7) ile alternatiflere ait Referans Noktası Yaklaşımı sıralaması elde edilir.

$$\min_j \{ \max_i (|w_i r_i - v_{ij}|) \} \quad (4.7)$$

4.2.3. Tam çarpım formu

Moora metoduna daha sonra eklenerek MultiMoora metodunun oluşturulmasında yararlanılan Tam Çarpım Formu metodunda, Moora metodu içerisinde yer alan Oran Metodu ve Referans Noktası Yaklaşımı'nın aksine başlangıç matrisi normalizasyon

işlemi uygulanmadan kullanılmaktadır. Tam Çarpım Formu uygulanırken aşağıda sıralanan işlem basamakları takip edilmektedir:

1.Adım: Tam Çarpım Formu, işlem basamakları başlangıç matrisinin kullanımı ile başlamaktadır. Başlangıç matrisine normalizasyon işlemi uygulanmadan, yalnızca ağırlıklandırma işlemi uygulanır. Ağırlıklandırma işlemi, matris değerlerinin x_{ij} bulundukları sütunlardaki kriterlere ait kriter ağırlığı w_i kadar kuvvetlerinin alınması (4.8) şeklinde gerçekleştirilir.

$$x_{ij}^{w_i} \quad (4.8)$$

2.Adım: Ağırlıklandırma işlemi ile ağırlıklandırılmış değerlerden yeni bir matris elde edilir. Bu matriste yer alan her bir alternatif için bulunduğu satırdaki maksimum kriter değerleri çarpımı A_j ve minimum kriter değerleri çarpımı B_j ayrı ayrı hesaplanır (4.9).

$$A_j = \prod_{i=1}^g x_{ij}^{w_i} \quad , \quad B_j = \prod_{i=g+1}^n x_{ij}^{w_i} \quad (4.9)$$

A_j = her bir alternatif için maksimum sütun değerleri çarpımı

B_j = her bir alternatif için minimum sütun değerleri çarpımı

3.Adım: Her bir alternatif için hesaplanan A_j ve B_j değerlerinin oranlanması ile U_j değeri elde edilir (4.10). Elde edilen bu değer büyükten küçüğe doğru sıralanması ile alternatiflere ait Tam Çarpım Formu sıralaması elde edilir.

$$U_j = A_j/B_j \quad (4.10)$$

U_j = alternatif j 'nin genel faydası

4.2.4. Sıra baskınlık teorisi

Oran Metodu, Referans Noktası Yaklaşımı ve Tam Çarpım Formu'nun uygulanması ile üç farklı alternatif sıralaması elde edilir. Bu sıralamaların birbirleri ile kıyaslanması işlemi Sıra Baskınlık Teorisi ile gerçekleştirilmektedir. Kıyaslama sonucu, üç

sıralamadan tek bir MultiMoora sıralaması elde edilir. Sıra baskınlık teorisi uygulanırken mutlak baskınlık, geçişkenlik, dengelilik, döngüsel akıl yürütme şeklinde ana başlıklar halinde oluşturulan kurallardan yararlanılmaktadır (Brauers ve Zavadskas, 2012; Türe ve ark., 2017). Bu kurallar aşağıdaki gibidir:

Mutlak Baskınlık: Eğer üç farklı teknikten elde edilen alternatif sıralama değerleri sırasıyla (2-2-2) ise, bu kesinlikle baskındır.

Genel Baskınlık: $(k < l < f < z)$ baskınlıkları verildiğinde, $(z - k - k)$, $(f - l - l)$ 'ye; $(k - z - k)$, $(l - f - l)$ 'ye ve $(k - k - z)$, $(l - l - f)$ 'ye genellikle baskındır.

Geçişkenlik: Eğer k , l 'ye ve l , f 'ye baskın ise k , f 'ye baskın olacaktır.

Örneğin $(k - k - k)$, $(l - l - l)$ 'ye tamamen baskındır.

Dengelilik: Örneğin; 2 alternatif de $(c - c - c)$ gibi bir sonuca sahipse mutlak dengelilik, Eğer $(4 - c - 11)$ ve $(6 - c - 7)$ gibi 3 olaydan 2'sinde dengelilik varsa, kısmi dengelilik durumu söz konusudur.

Yukarıda belirtilen sınıflandırmaların dışında çelişkili durumlar meydana gelebilmektedir. Örneğin; A (3-11-5) alternatifi B (5-7-6) alternatifine, B (5-7-6) alternatifi C (6-10-4) alternatifine, C (6-10-4) alternatifi ise A (3-11-5) alternatifine genellikle baskındır. Döngüsel akıl yürütme olarak da adlandırılan böyle bir durumda, 3 alternatife de aynı sıra verilmektedir.

BÖLÜM 5. UYGULAMA

5.1. Önerilen Hibrit Modeller

Genetik algoritmalarından, çeşitli problemlere çözüm getirmek amacıyla yararlanılan birçok çalışma mevcuttur. Bu çalışmada genetik algoritmalar, çok iş ve çok tezgahtan oluşan çok amaçlı akış tipi çizelgeleme probleminin çözümünde kullanılmıştır. Algoritmanın, birden fazla kriteri birlikte değerlendiren ve çok amaçlı bir yapısı olan probleme çözüm getirmede performansını arttırmak amacıyla, genetik algoritmalara çok kriterli karar verme yöntemlerinden biri olan MultiMoora metodu entegre edilmiştir. MultiMoora metodu genetik algoritmaların öncelikle yalnızca seçim adımına entegre edilerek farklı varyasyonlar ile MBGA-1 olarak gruplandırılan MBGA-1-A, MBGA-1-B ve MBGA-1-C hibrit algoritmaları geliştirilmiştir. Daha sonra MultiMoora metodu, genetik algoritmaların çaprazlama ve mutasyon işlemleri sonucu üretilen yeni bireylerin popülasyona aktarılması amacıyla eski bireylerden bazılarının elendiği, eleme adımına entegre edilerek MBGA-2 algoritması oluşturulmuştur. Son olarak, MultiMoora metodu genetik algoritmaların hem seçim hem de eleme adımlarına birlikte entegre edilerek bir hibrit algoritma olan, MBGA-3 algoritması geliştirilmiştir. Oluşturulan 5 hibrit algoritmanın performansı seçim stratejisi olarak Turnuva Seçimi, eleme stratejisi olarak En Kötü Olanı Eleme stratejisinin kullanıldığı çok amaçlı genetik algoritma (MOGA) ile karşılaştırılarak değerlendirilmiştir.

5.1.1. Genetik algoritmaların seçim adımında MultiMoora metodunun uygulanması: MultiMoora tabanlı genetik algoritma-1 (MBGA-1)

Genetik algoritmaların çaprazlama işlemi uygulanacak bireylerin seçildiği seçim adımına, farklı varyasyonlar ile çok kriterli karar verme yöntemlerinden biri olan

MultiMoora entegre edilmiştir. Metotların 3 farklı varyasyon ile hibritlenmesi sonucu oluşturulan ve MBGA-1 olarak gruplandırılan hibrit algoritmalar tasarlanmıştır. MBGA-1-A, MBGA-1-B, MBGA-1-C olarak adlandırılan 3 hibrit algoritma ve bu algoritmalara ait işlem adımları bu bölümde detaylandırılmıştır.

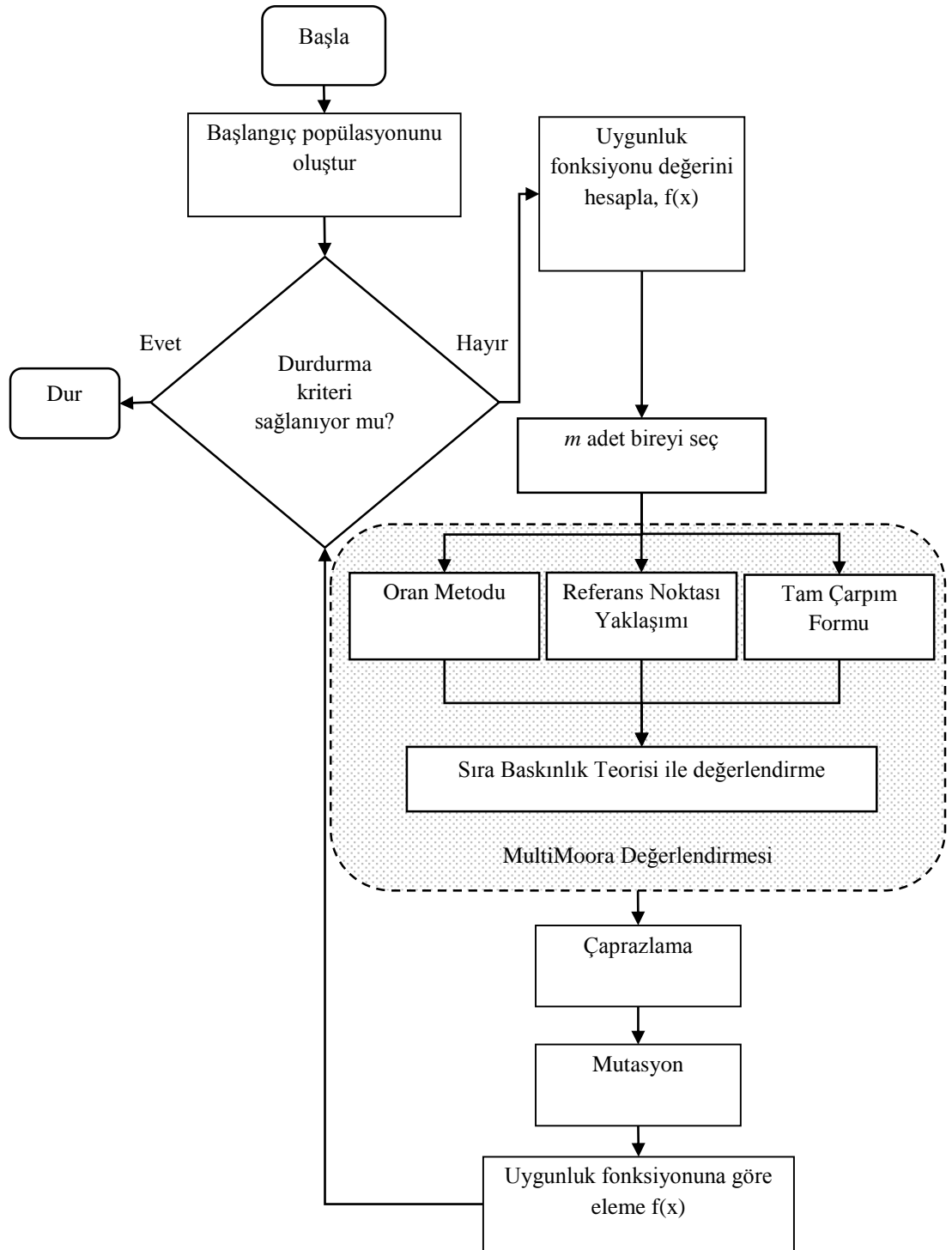
5.1.1.1. MBGA-1-A hibrit algoritması

Genetik algoritmaların seçim adımına MultiMoora'nın entegre edilmesi ile oluşturulan MBGA-1-A hibrit algoritmasına ait işlem adımları aşağıda sıralanmaktadır:

1. Adım: Başlangıç çözümlerinin kromozomlar halinde kodlanması ile N adet kromozomdan meydana gelen bir başlangıç popülasyonu oluşturulur.
2. Adım: Başlangıç popülasyonunda yer alan tüm bireylerin, uygunluk fonksiyonunu ifade eden denklem (4.1) kullanılarak uygunluk değeri hesaplanır.
3. Adım: En iyi uygunluk değerine sahip m adet birey MultiMoora işlemi uygulanmak üzere seçilir.
4. Adım: Seçilen m adet bireye maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve zamanında biten iş sayısı NTJ kriterleri açısından MultiMoora metodu uygulanır. MultiMoora metodunda Moora-Oran Metodu, Moora-Referans Noktası Yaklaşımı ve Tam Çarpım Formu metotlarının uygulanmasının ardından, bu üç metot ile elde edilen sıralamaların Sıra Baskınlık Teorisi ile kıyaslanması ve MultiMoora sıralamasının elde edilmesi işlemleri uygulanır.
5. Adım: Elde edilen MultiMoora sıralamasında yer alan ilk iki birey çaprazlama işlemi uygulanmak üzere seçilir. Uygulanan adımların belirlenen çaprazlama oranınca tekrarlanması ile, çaprazlama işlemi için ihtiyaç duyulan tüm çapraz çiftlerinin oluşturulması sağlanır.

6. Adım: Çaprazlama ve mutasyon işlemleri uygulanarak yeni bireyler oluşturulur.
7. Adım: Oluşturulan yeni bireyler, eleme stratejisi ile elenen eski bireylerin yerini alacak şekilde popülasyona aktarılır.
8. Adım: Belirlenen durdurma kriteri sağlanana kadar uygulama sürdürülür.

Geliştirilen MBGA-1-A hibrit algoritmasının uygulanması sırasında takip edilecek adımlar, Şekil 5.1.'de yer alan şema ile gösterilmiştir.



Şekil 5.1. Önerilen MBGA-1-A hibrit algoritması akış şeması

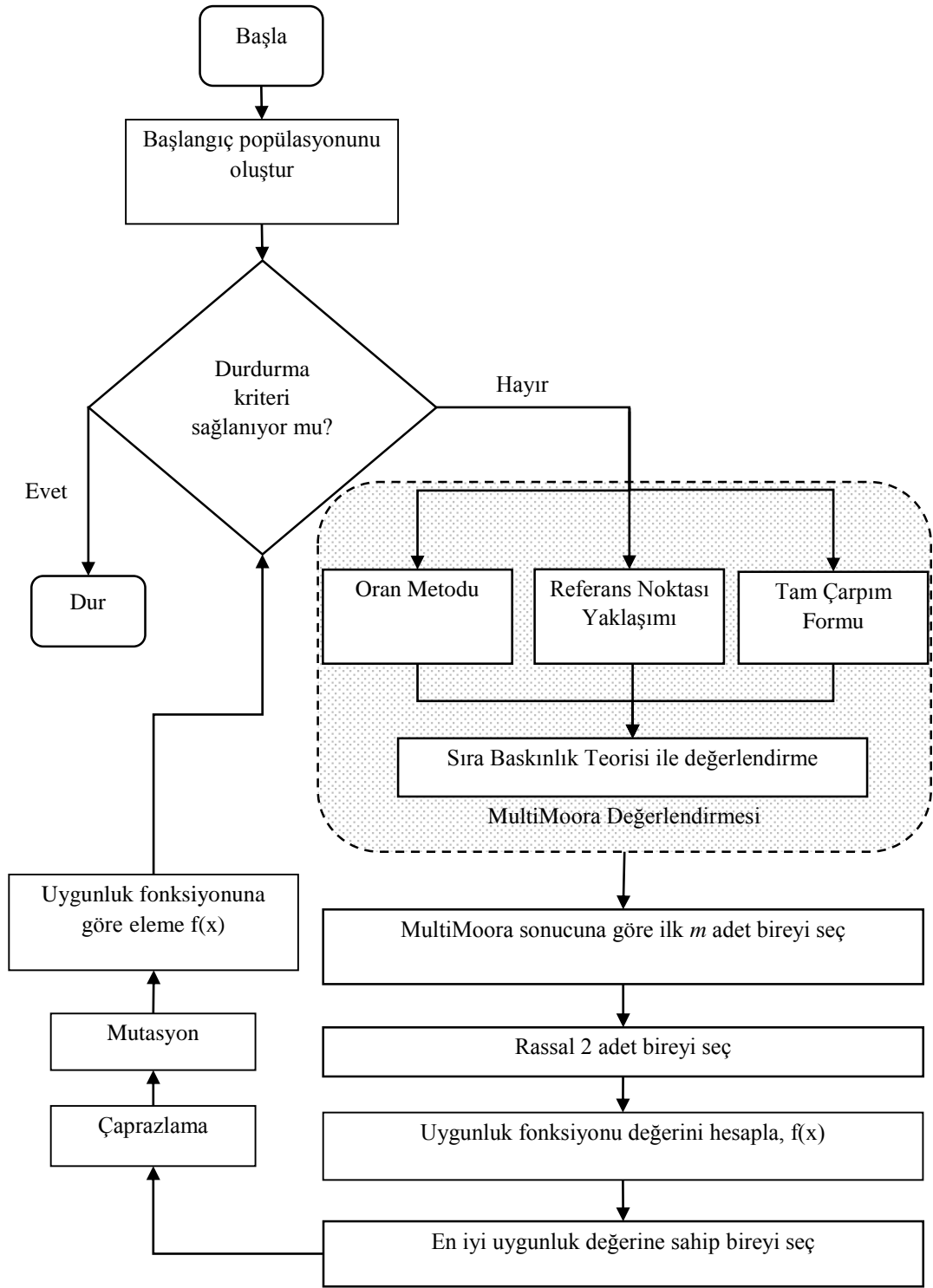
5.1.1.2. MBGA-1-B hibrit algoritması

Genetik algoritmaların seçim adımında MultiMoora ile değerlendirme sağlayan MBGA-1-B hibrit algoritmasının uygulanması sırasında takip edilmesi gereken işlemler aşağıda sıralanmıştır:

1. Adım: Başlangıç kromozomlarından oluşan N adetlik bireye sahip bir başlangıç popülasyonu oluşturulur.
2. Adım: Başlangıç popülasyonunda yer alan tüm bireylere maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve zamanında biten iş sayısı NTJ kriterleri açısından MultiMoora metodu uygulanarak, bireylere ait bir MultiMoora sıralaması elde edilir.
3. Adım: MultiMoora sıralamasına göre ilk m adet birey popülasyondaki bireyler arasından seçilir.
4. Adım: Seçilen m adet birey arasından rassal olarak 2 birey seçilir.
5. Adım: Seçilen 2 bireyin uygunluk fonksiyonu ile uygunluk değeri hesaplanır.
6. Adım: Uygunluk değeri hesaplanan bu iki bireyden, en iyi uygunluk değerine sahip olan birey çaprazlama işlemi uygulanmak üzere seçilir. Çaprazlama oranınca devam edilerek çaprazlama için ihtiyaç duyulan kadar birey aynı şekilde seçilir.
7. Adım: Seçilen bireylere çaprazlama ve mutasyon işlemleri uygulanarak yeni jenerasyon bireyler oluşturulur.
8. Adım: Oluşturulan yeni bireyler, popülasyonda başlangıçta yer alan bireylerin uygun eleme stratejisi ile elenmesi sonucu popülasyona aktarılır.

9. Adım: Algoritma için belirlenen durdurma kriteri sağlanıncaya kadar algoritma adımları tekrarlanır.

Önerilen MBGA-1-B hibrit algoritmasının uygulama adımları Şekil 5.2.'deki şema ile gösterilmiştir.



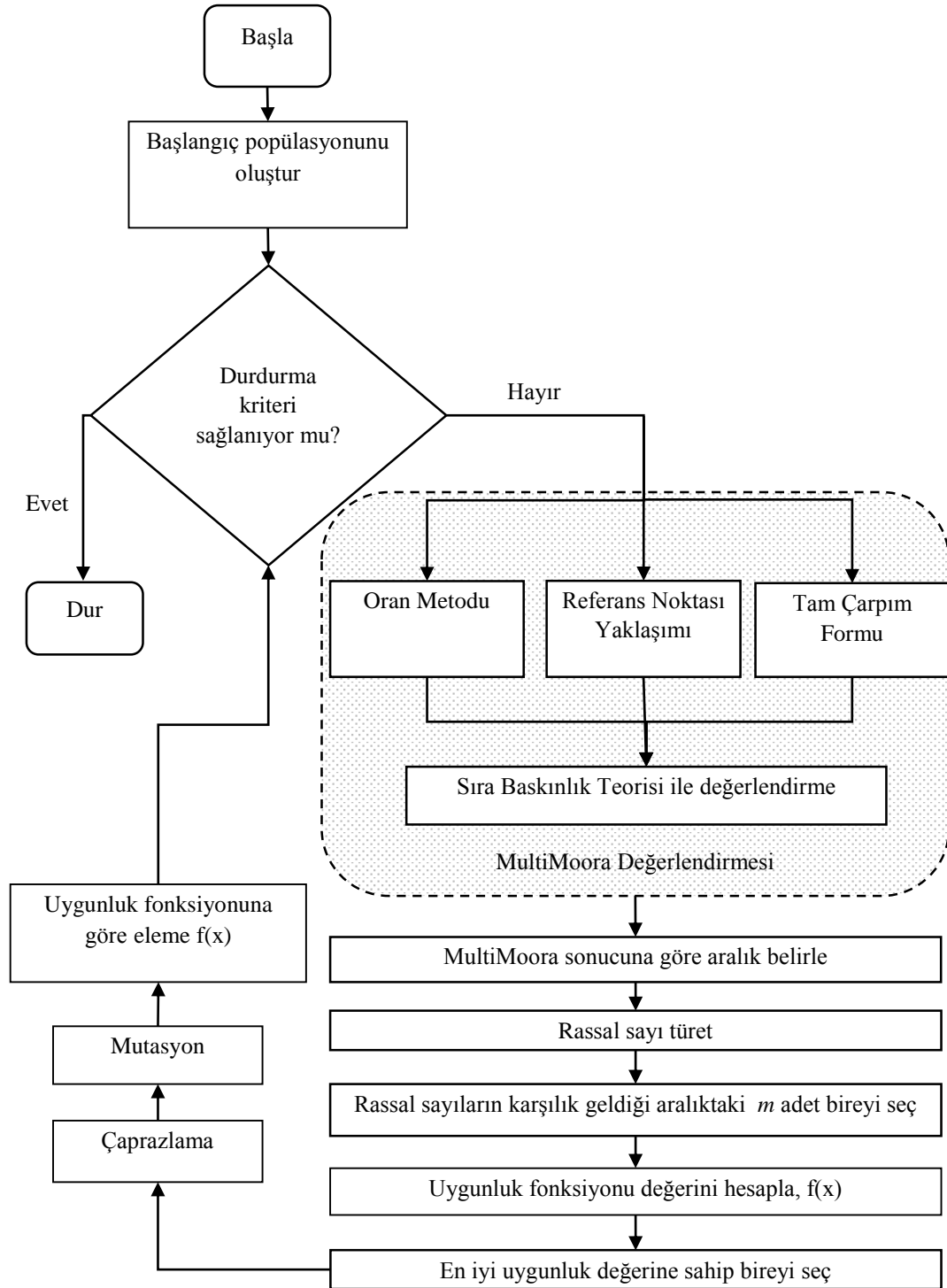
Şekil 5.2. Önerilen MBGA-1-B hibrit algoritması akış şeması

5.1.1.3. MBGA-1-C hibrit algoritması

MultiMoora metodu ile genetik algoritmaların seçim adımında değerlendirme sağlamak amacıyla önerilen, MBGA-1-C hibrit algoritmasına ait işlem adımları aşağıda sıralanmıştır.

1. Adım: N adet kromozomdan meydana gelen başlangıç popülasyonu oluşturulur.
2. Adım: Başlangıç popülasyonundaki kromozomlara değerlendirmede maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve zamanında biten iş sayısı NTJ kriterleri kullanılarak MultiMoora metodu uygulanır.
3. Adım: MultiMoora sıralamaları oranınca bireylere $[0,100]$ aralığında alan sağlanır.
4. Adım: Rassal sayılar türetilerek, bu rassal sayıların karşılık geldiği alandaki bireyler seçilir. Bu işlem ile m adet bireyin seçilmesi sağlanır.
5. Adım: Seçilen m adet bireyin uygunluk fonksiyonu kullanılarak uygunluk değerleri hesaplanır.
6. Adım: Uygunluk değerleri hesaplanan m adet birey arasından, en iyi uygunluk değerine sahip olan birey çaprazlama işlemi uygulanmak üzere seçilir. Çaprazlama oranınca birey seçilerek, çaprazlama için ihtiyaç duyulan birey sayısı tamamlanır.
7. Adım: Seçilen bireylere çaprazlama ve mutasyon işlemleri uygulanır. Böylece yeni bireyler oluşturulur.
8. Adım: Popülasyonda önceden yer alan bireyler arasından, seçilen eleme stratejisi doğrultusunda, yeni jenerasyon birey sayısı kadar birey elenir ve elenen bireylerin yerine yeni bireyler popülasyona aktarılır.

9. Adım: Durdurma kriteri sağlanıncaya kadar algoritma adımları tekrarlanır. Geliştirilen MBGA-1-C hibrit yapısının uygulanması sırasında takip edilen adımlar Şekil 5.3.'de yer alan şema ile gösterilmiştir.



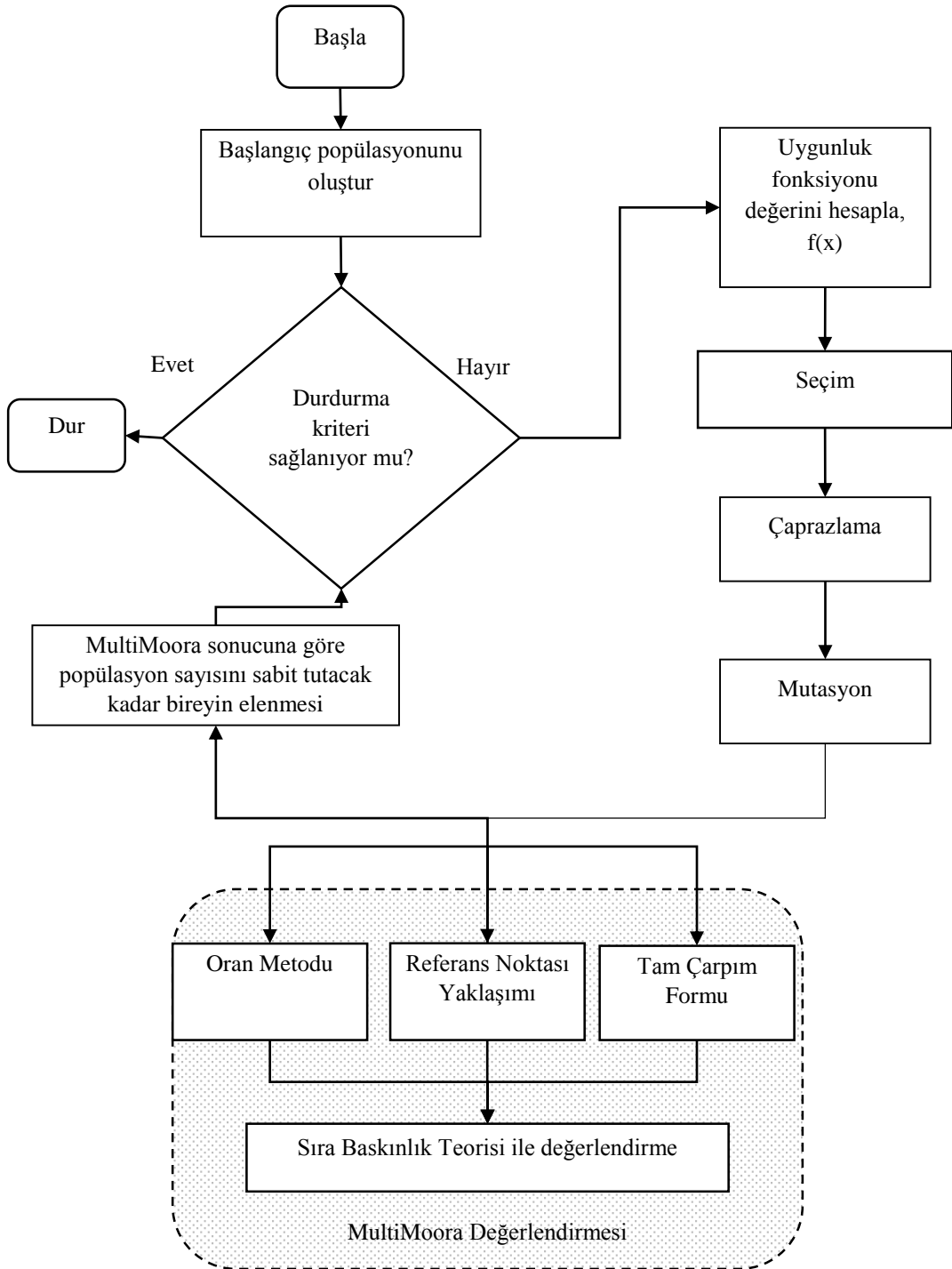
Şekil 5.3. Önerilen MBGA-1-C hibrit algoritması akış şeması

5.1.2. Genetik algoritmaların eleme adımında MultiMoora metodunun uygulanması: MBGA-2 hibrit algoritması

Genetik algoritmaların çaprazlama ve mutasyon adımlarının uygulanması ile elde edilen yeni jenerasyon bireylerin popülasyona aktarılması sırasında, yeni bireylerin popülasyonda yer alan bireylerden hangilerinin yerine aktarılacağı kararının verildiği adım eleme adımıdır. Bu adıma çok kriterli karar verme yöntemlerinden biri olan MultiMoora metodu entegre edilerek MBGA-2 hibrit algoritması geliştirilmiştir. MultiMoora metodu ile popülasyonda yer alan bireyler arasından yeni bireylere yer sağlamak amacıyla maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve zamanında biten iş sayısı NTJ kriterlerinin birlikte değerlendirildiği çok amaçlı bir eleme hedeflenmiştir. Geliştirilen MBGA-2 algoritmasının uygulama adımları aşağıdaki gibidir:

1. Adım: Önerilen hibrit yapının ilk adımında başlangıç popülasyonunun oluşturulması yer almaktadır.
2. Adım: Başlangıç popülasyonuna sırasıyla seçim, çaprazlama ve mutasyon işlemleri uygulanarak yeni jenerasyon bireyler elde edilir.
3. Adım: Popülasyonda önceden yer alan bireylere MultiMoora metodu işlemleri uygulanır. MultiMoora metodu uygulanırken Moora-Oran Metodu, Moora-Referans Noktası Yaklaşımı ve Tam Çarpım Formu metotları ile sıralamalar elde edilerek bu metotlara ait sıralamaların Sıra Baskınlık Teorisi ile değerlendirilmesi ve MultiMoora sıralamasının elde edilmesi adımları takip edilir.
4. Adım: MultiMoora sonucuna göre en alt sırada yer alan kromozomlardan, yeni jenerasyon birey sayısı kadarı popülasyondan elenir.
5. Adım: Elenen eski bireylerin yerine, yeni jenerasyon bireyler popülasyona aktarılır.
6. Adım: Durdurma kriteri sağlanıncaya kadar algoritma adımları tekrarlanır.

Geliştirilen MBGA-2 hibrit modelinin uygulanması sırasında takip edilmesi gereken adımlar, Şekil 5.4.'te yer alan şema ile gösterilmiştir.



Şekil 5.4. Önerilen MBGA-2 hibrit algoritması akış şeması

5.1.3. Genetik algoritmaların seçim ve eleme adımında MultiMoora metodunun uygulanması: MBGA-3 hibrit algoritması

Genetik algoritmaların çaprazlama işlemi uygulanacak kromozomların seçildiği seçim adımına ve aynı zamanda yeni jenerasyon bireylerin popülasyona aktarılması sırasında popülasyonda önceden yer alan bireylerin elendiği eleme adımına, çok kriterli karar verme yöntemlerinden biri olan MultiMoora entegre edilmiştir. Bu işlem sonucu, hibrit bir yapı olan MBGA-3 algoritması oluşturulmuştur. Bu hibrit algoritma ile genetik algoritmaların hem seçim adımında hem de eleme adımında çok amaçlı değerlendirme sağlanarak algoritma performansının artırılması hedeflenmiştir. Algoritma adımları aşağıdaki gibidir:

1. Adım: Başlangıç popülasyonu oluşturulur.
2. Adım: Başlangıç popülasyonu bireylerinin tamamına MultiMoora metodu işlem adımları uygulanır ve popülasyondaki bireyler için MultiMoora sıralaması elde edilir.
3. Adım: MultiMoora sonucu en iyi olan m adet birey seçilir.
4. Adım: Seçilen m adet birey arasından rassal olarak 2 birey seçilir.
5. Adım: Seçilen 2 bireyin uygunluk fonksiyonu ile uygunluk değeri hesaplanır.
6. Adım: Uygunluk değeri en iyi olan birey çaprazlama işlemi için seçilir.
7. Adım: Çaprazlama oranınca aynı seçim sürecinin gerçekleştirilmesi ile çaprazlama işlemi için gereken tüm bireylerin seçilmesi sağlanır.
8. Adım: Çaprazlama ve mutasyon işlemleri uygulanarak yeni jenerasyon bireyler elde edilir.

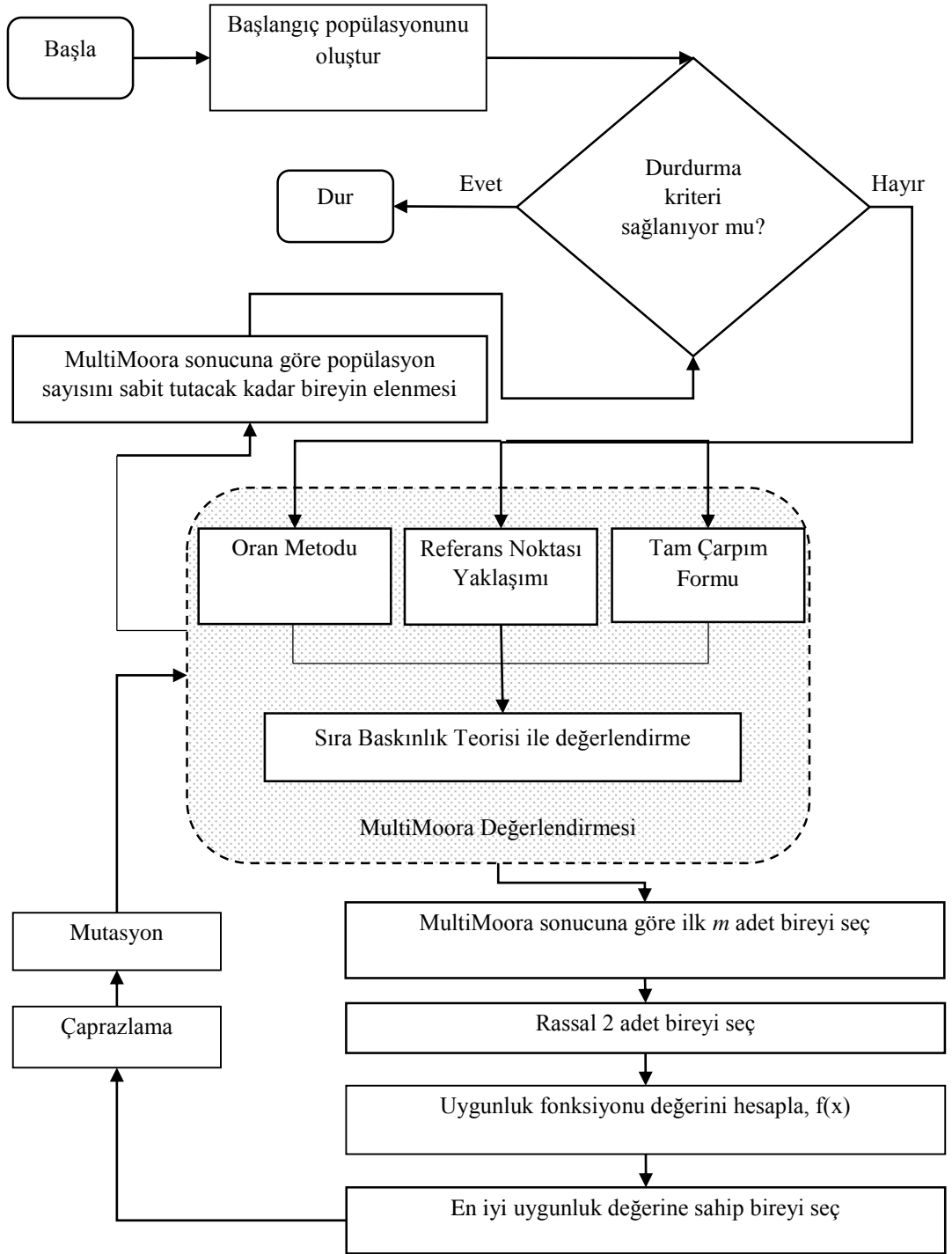
9. Adım: Popülasyonda önceden var olan bireylerin tamamına MultiMoora işlemi uygulanır.

10. Adım: MultiMoora işlemi ile bireylere ait sıralama elde edilir. Elde edilen sıralamada en kötü sıraya sahip olan, yeni jenerasyon birey sayısı kadar eski birey popülasyondan elenir.

11. Adım: Popülasyondan elenen eski bireylerin yerine, yeni jenerasyon bireyler popülasyona aktarılır.

12. Adım: Durdurma kriteri sağlanıncaya kadar algoritma adımları tekrarlanır.

MBGA-3 algoritmasının uygulanması sırasında takip edilmesi gereken adımlar Şekil 5.5.'teki şema ile gösterilmiştir.



Şekil 5.5. Önerilen MBGA-3 hibrit algoritması akış şeması

5.2. Problemin Yapısı

Çalışmada, geliştirilen hibrit modeller, farklı iş sayıları ile oluşturulan 4 farklı çok amaçlı akış tipi çizelgeleme problemi üzerinde yapılan uygulamalar ile test edilmiştir. Belirlenen test problemleri, maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve geciken iş sayısı NLJ olmak üzere 5 kriterin birlikte optimizasyonunun hedeflendiği çok amaçlı akış tipi çizelgeleme problemleridir. Bu problemler sırasıyla 15 iş, 25 iş, 50 iş ve 100 işin akış tipi çizelgelenmesini içermektedir. İşlerin, 10 farklı tezgahta önceden belirlenen işlem süreleri ve rotaları doğrultusunda birbirlerine göre işlem önceliklerinin belirlenmesi hedeflenmektedir.

Problemlerde birlikte optimizasyonu hedeflenen 5 kriterin birbirlerine göre önem düzeyini ifade eden kriter ağırlıklarının, her bir kriter için 0,2 olmak üzere toplamı 1 olacak şekilde eşit olduğu kabul edilmiştir. 4 test problemi için de 40 kromozomdan oluşan başlangıç popülasyonları ile uygulama adımları başlatılmıştır. Her bir iş sayısı için, geliştirilen hibrit modeller çok amaçlı genetik algoritma (MOGA) ile aynı başlangıç popülasyonu kullanılarak uygulanmıştır. MOGA için algoritma, seçim stratejisi olarak turnuva seçimi ve eleme stratejisi olarak en kötü olanı eleme stratejileri kullanılarak oluşturulmuştur. Geliştirilen hibrit algoritmalar, MOGA ile aynı çaprazlama ve mutasyon metotları ile uygulanmıştır. Çaprazlama işlemi doğrusal sıralı çaprazlamanın farklı bir şekilde uygulanması, mutasyon işlemi ise keyfi üç geni değiştirme mutasyonunun uygulanması ile gerçekleştirilmiştir. Geliştirilen hibrit modeller içerisinde MultiMoora metodu uygulanırken, geciken iş sayısı NLJ kriteri yerine zamanında biten iş sayısı NTJ kriteri maksimum bir kriter olarak değerlendirmeye alınmıştır. Uygulama ile elde edilen sonuçların MOGA ile kıyaslanabilmesi amacıyla, zamanında biten iş sayısı değerleri, geciken iş sayısı değerlerine dönüştürülerek kullanılmıştır.

Geliştirilen 5 algoritma ve MOGA, Excel programı üzerinde makro programlama ile kodlanmıştır. Kodlanan algoritmalar, test problemleri üzerinde 200 jenerasyon çalıştırılmış, elde edilen sonuçlar ortalama uygunluk fonksiyonu $f(x)_{ort}$ ve minimum

uygunluk fonksiyonu $f(x)_{min}$ değerlerinde meydana gelen değişimin grafikler üzerinde izlenmesi ile kıyaslanmıştır. Değerlendirme sırasında, maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} , geciken iş sayısı NLJ kriterleri ve $f(x)_{min}$ değerlerinde her jenerasyonda iş sayıları bazında meydana gelen değişimin yer aldığı tablolardan da yararlanılmıştır.

5.3. Algoritmaların Program Üzerinde Uygulanması

Çalışmada MBGA-1-A, MBGA-1-B, MBGA-1-C, MBGA-2, MBGA-3 ve MOGA algoritmaları 15, 25, 50 ve 100 iş sayılarından oluşan 4 farklı test problemi üzerinde uygulanmıştır. Uygulama, Excel’de makro programlama ile algoritmaların kodlanmasıyla oluşturulan, program üzerinde gerçekleştirilmiştir. Akış tipi çizelgeleme problemlerinde, çizelgelenecek işlerin 10 farklı tezgahta işlenmesi için geçen süre ve işlerin teslim süresi bilgileri programa eklenmiştir. Başlangıç popülasyonu büyüklüğü 40 kromozom olarak belirlenmiştir. Algoritmaların durdurma kriterini temsil eden, jenerasyon sayısı değeri 200 olarak belirlenmiş ve bu değer program üzerinde ilgili alana yazılmıştır. Çaprazlama işleminin, kromozomların hangi genleri arasında yer alan bölümüne uygulanacağı, kromozom üzerinde çizelgelenen iş sayısı doğrultusunda belirlenmiştir. Akış tipi çizelgeleme problemlerinde birlikte minimizasyonu hedeflenen 5 kritere ait kriter ağırlıkları, programın oluşturulan ilgili alanında belirtilmiştir. MOGA’nın seçim adımında uygulanan turnuva seçimi için gerekli olan turnuva genişliği değeri belirlenmiştir. Geliştirilen hibrit algoritmaların uygulanması sırasında kullanılan m değeri, program üzerinde ilgili alanda belirtilmiştir. Başlangıç değerlerinin belirlenmesinin ardından Şekil 5.6.’da programa ait ekran görüntüsünde yer alan ‘Başlangıç Popülasyonunu Güncelle’ butonu ile, popülasyon büyüklüğü kadar rassal bireyden meydana gelen başlangıç popülasyonunun oluşturulması sağlanmıştır. Algoritmaların çalışması için gerekli olan düzenlemelerin ardından, programın ‘Algoritmalar’ bölümünden çalıştırılacak algoritma seçilmiş ve ‘Algoritmayı Çalıştır’ butonu ile seçili algoritmanın çalışması sağlanmıştır. Algoritmanın çalışması süresince işlem yapılan jenerasyon sayısı, oluşturulan jenerasyon takip sayacı ile takip edilmiştir. Uygulama sonucu, çalıştırılan

algoritmaya ait 200 jenerasyon boyunca elde edilen $f(x)_{ort}$, $f(x)_{min}$ değerleri ve 5 kritere ait değerler programda listelenmiştir. Algoritmaların performansı, elde edilen bu sonuçlar çerçevesinde değerlendirilmiştir.

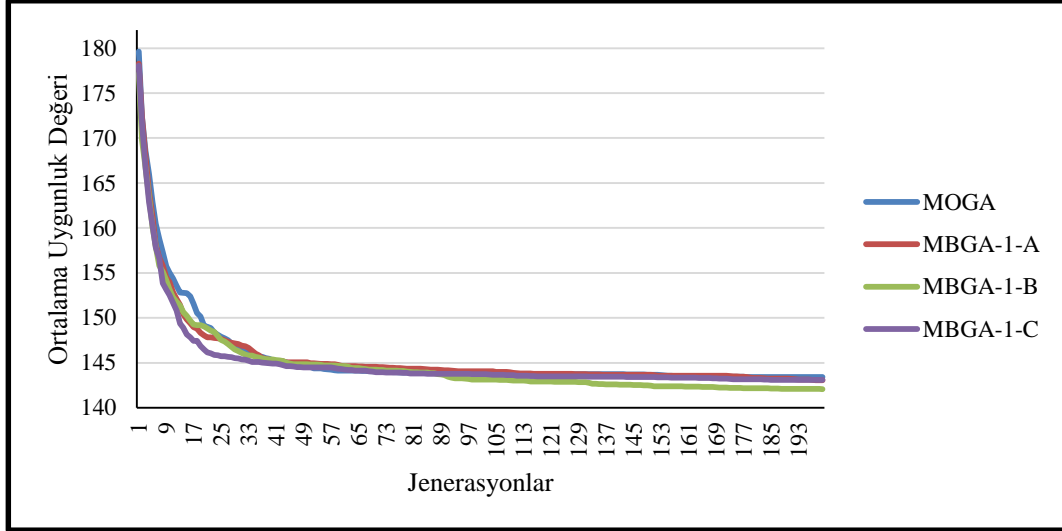
M4																												
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W					
1															Kriter Ağırlıkları				Algoritma				Turnuva genişliği		5			
2	Popülasyon büyüklüğü:				40	Jenerasyon takip sayısı:				200	Jenerasyon sayısı:				200	C_{max}	F	T_{max}	T	NLI	<input checked="" type="radio"/> MGA				m değeri:	5		
3	İş sayısı:				15	Tezgaah sayısı:				10	Çaprazlama aralığı:				3	12	0,2	0,2	0,2	0,2	<input type="radio"/> MGA-1-A					5		
																					<input type="radio"/> MGA-1-B							
																					<input type="radio"/> MGA-1-C							
																					<input type="radio"/> MGA-2							
																					<input type="radio"/> MGA-3							
	Başlangıç Popülasyonunu Güncelle														Algoritmayı Çalıştır													
4																												
5	10	9	2	5	4	15	8	6	13	1	7	14	3	11	12													
6	4	11	12	9	5	6	14	1	13	3	15	2	8	10	7													
7	5	14	13	12	10	7	15	9	2	6	11	1	4	8	3													
8	12	5	8	10	13	15	11	7	3	6	14	9	1	4	2													
9	11	7	5	9	2	8	10	3	12	14	4	1	15	13	6													
10	5	2	7	10	14	12	8	6	9	13	1	3	4	15	11													
11	9	8	4	5	14	1	2	15	7	6	13	11	12	3	10													
12	3	1	4	5	9	2	14	7	13	15	8	11	12	6	10													
13	4	14	10	3	1	6	7	2	12	5	11	9	15	13	8													
14	11	8	15	12	5	7	6	10	1	9	3	13	2	14	4													
15	1	10	14	4	7	3	13	15	6	8	12	9	11	5	2													
16	10	14	15	4	1	6	7	5	8	13	3	14	9	3	13													

Şekil 5.6. Algoritmaların çok amaçlı akış tipi çizelgeleme probleminde uygulandığı programa ait ekran görüntüsü

BÖLÜM 6. SONUÇ VE DEĞERLENDİRME

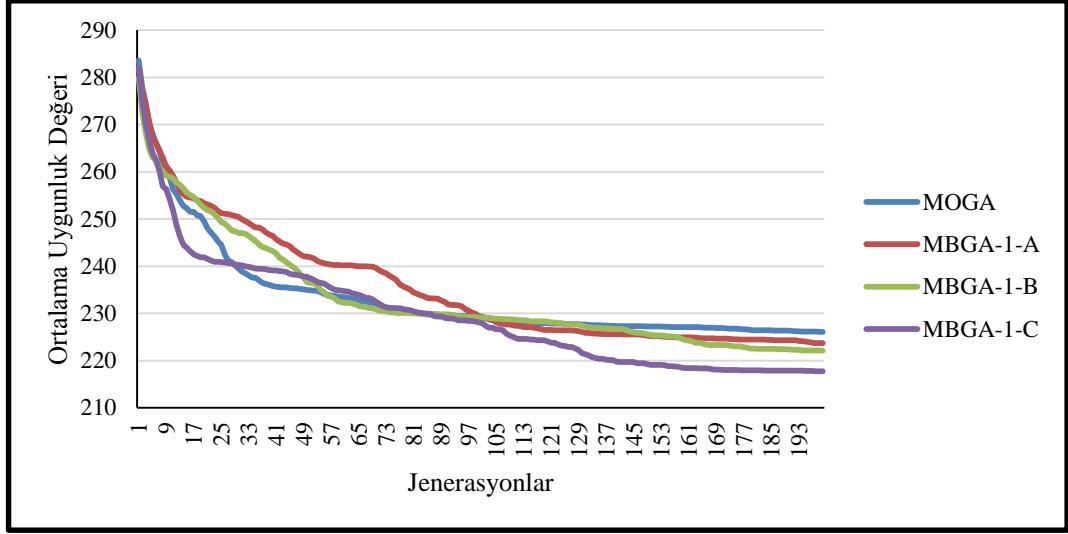
MultiMoora metodunun genetik algoritmaların seçim, eleme ve hem seçim hem de eleme adımına eklenmesi ile 3 grup halinde geliştirilen 5 hibrit algoritma ve çok amaçlı genetik algoritmalar, 4 farklı iş sayısından oluşan çok amaçlı akış tipi çizelgeleme problemleri üzerinde uygulanmıştır. Uygulama sonucu uygunluk fonksiyonu $f(x)$, maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve geciken iş sayısı NLJ kriterleri açısından algoritmaların performansları değerlendirilmiştir. Değerlendirmede, 3 grup olarak geliştirilen algoritmalara ait $f(x)_{ort}$ ve $f(x)_{min}$ değerlerinde jenerasyonlar boyunca meydana gelen değişiminin yer aldığı sonuçların, grup bazında MOGA sonuçları ile karşılaştırıldığı grafiklerden yararlanılmıştır. Aynı zamanda geliştirilen 5 algoritmanın sonuçlarının, MOGA ile aynı grafikler üzerinde 15 iş, 25 iş, 50 iş ve 100 iş sayıları bazında $f(x)_{ort}$ ve $f(x)_{min}$ değerleri açısından kıyaslanması sağlanmıştır. Jenerasyonlar sonunda her test problemi açısından $f(x)_{min}$ değerleri ve $f(x)_{min}$ değerlerinin elde edildiği kriter değerleri tablolar üzerinde gösterilerek, algoritmaların başarısı değerlendirilmiştir.

Şekil 6.1.'de yer alan grafik incelendiğinde 15 işten oluşan test problemi üzerinde yapılan uygulama sonucu, MBGA-1 algoritmalarının $f(x)_{ort}$ değerlerinde ilk jenerasyonlarda büyük oranda azalma sağladığı görülmektedir. Jenerasyon sayısının artışı ile birlikte, MBGA-1 algoritmaları MOGA ile yakın bir minimizasyon performansı göstermiştir. MBGA-1-C algoritması ilk jenerasyonlarda $f(x)_{ort}$ değerinin minimizasyonunda daha başarılı olmuştur. Son jenerasyonlar incelendiğinde MBGA-1-B algoritmasının diğer algoritmalarından daha düşük $f(x)_{ort}$ değerleri sağladığı tespit edilmiştir.



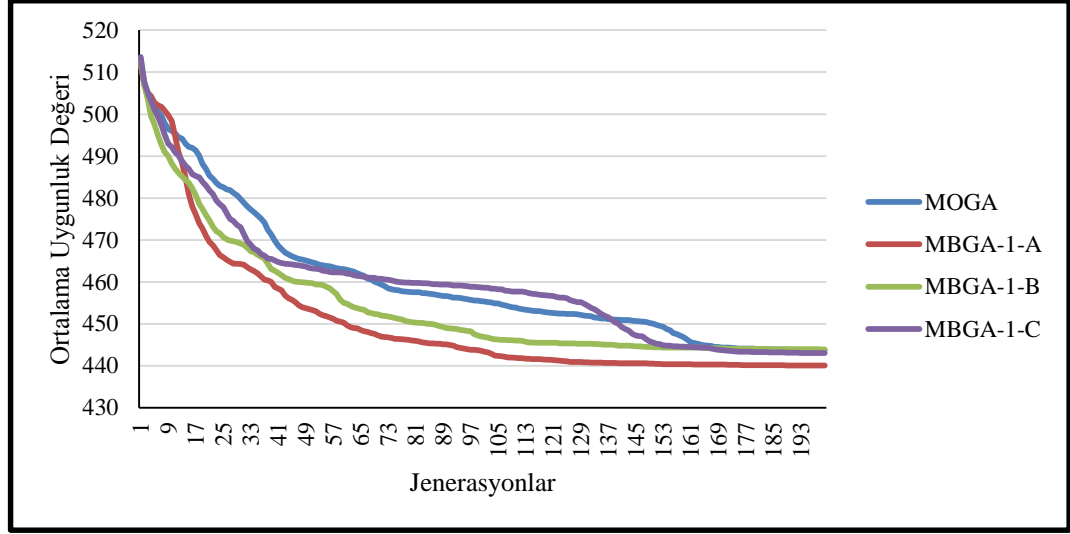
Şekil 6.1. 15 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

25 iş test problemi $f(x)_{ort}$ değerlerinin jenerasyonlardaki değişimine ait Şekil 6.2.'de yer alan grafikte, MBGA-1-C algoritmasının $f(x)_{ort}$ değerlerinde diğer algoritmalara kıyasla daha erken azalma sağladığı görülmektedir. İlk jenerasyonlarda başarılı oranda $f(x)_{ort}$ değerlerinde azalma sağlayan MOGA, ilerleyen jenerasyonlarda MBGA-1 algoritmalarının $f(x)_{ort}$ değerlerinde daha fazla minmizasyon sağlaması ile birlikte, diğer algoritmaların gerisinde kalmıştır. Son jenerasyonlar incelendiğinde, MBGA-1-C algoritmasının bu test problemi üzerinde $f(x)_{ort}$ değerlerinde daha fazla azalma sağladığı tespit edilmiştir.



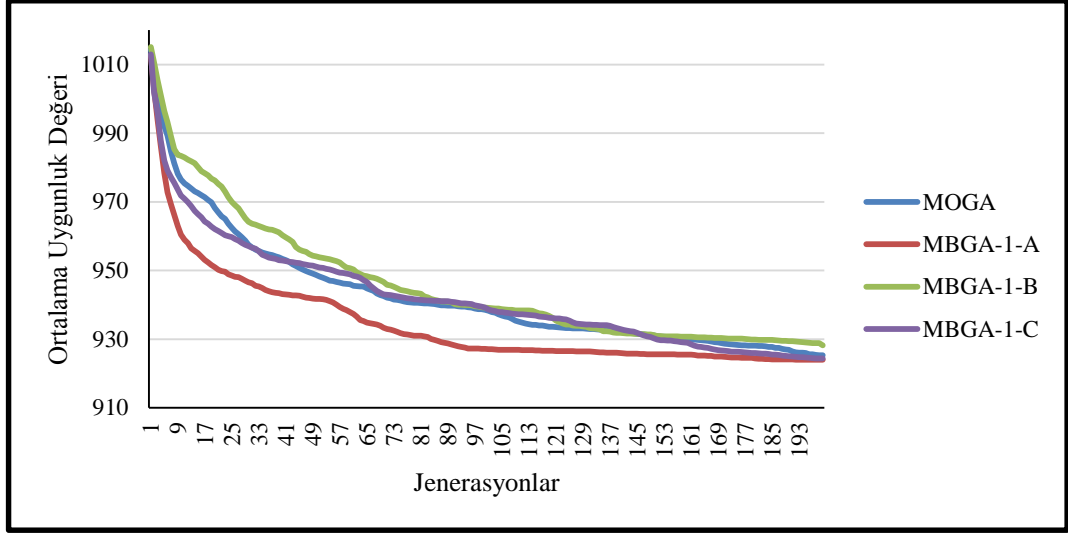
Şekil 6.2. 25 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

Şekil 6.3.'teki grafikte yer alan 50 iş problemi sonuçlarına göre, $f(x)_{ort}$ değerlerinde en hızlı azalmayı MBGA-1-A algoritmasının sağladığı görülmektedir. MBGA-1-A algoritması jenerasyonlar tamamlanıncaya kadar en yüksek performansı göstermiştir. MBGA-1-B algoritması, MBGA-1-A algoritmasına kıyasla daha yüksek $f(x)_{ort}$ değerleri elde etmiş, fakat MBGA-1-C ve MOGA'dan daha önceki jenerasyonlarda, daha düşük fonksiyon değerlerine ulaşmıştır. MBGA-1-A algoritması son jenerasyonlarda en düşük $f(x)_{ort}$ değerlerine ulaşmış, diğer 3 algoritma ise birbirlerine oldukça yakın $f(x)_{ort}$ değerleri elde edilmesini sağlamıştır.



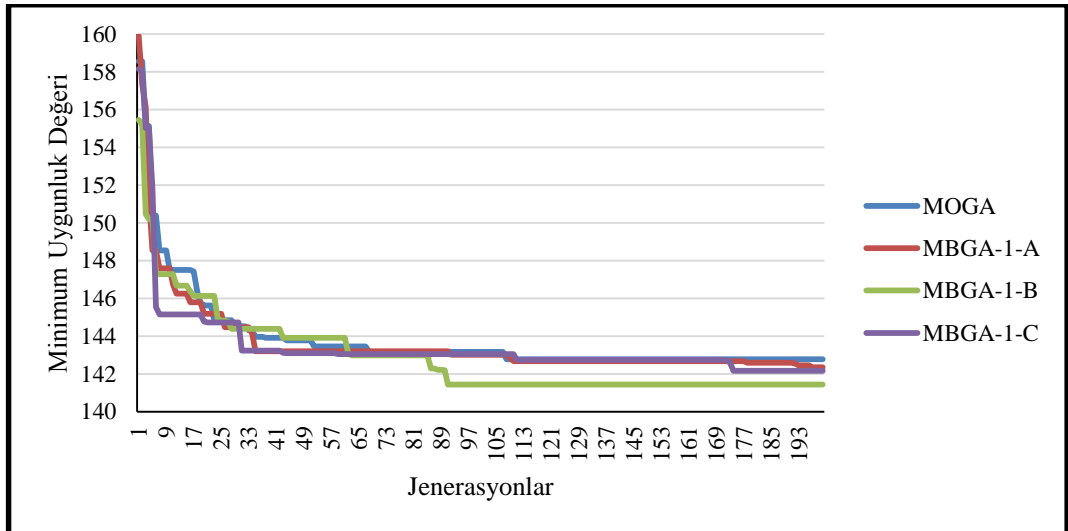
Şekil 6.3. 50 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

MBGA-1-A, 50 iş probleminde olduğu gibi 100 iş probleminde de Şekil 6.4.'te görüldüğü gibi $f(x)_{ort}$ değerlerinde, karşılaştırılan diğer algoritmalarından daha erken azalma sağlamıştır. MBGA-1-A algoritması jenerasyonlar boyunca diğer 3 algoritmayla oranla $f(x)_{ort}$ değerlerinde daha fazla minimizasyon ile grafik üzerinde daha aşağıda bir eğri oluşturmuştur. MBGA-1-B, MBGA-1-C ve MOGA $f(x)_{ort}$ değerlerinde yaklaşık aynı oranlarda azalma meydana getirmiştir. Son jenerasyonlar incelendiğinde MBGA-1-C algoritmasının $f(x)_{ort}$ değerlerinde MBGA-1-B ve MOGA'ya oranla daha fazla azalma sağladığı tespit edilmiştir.



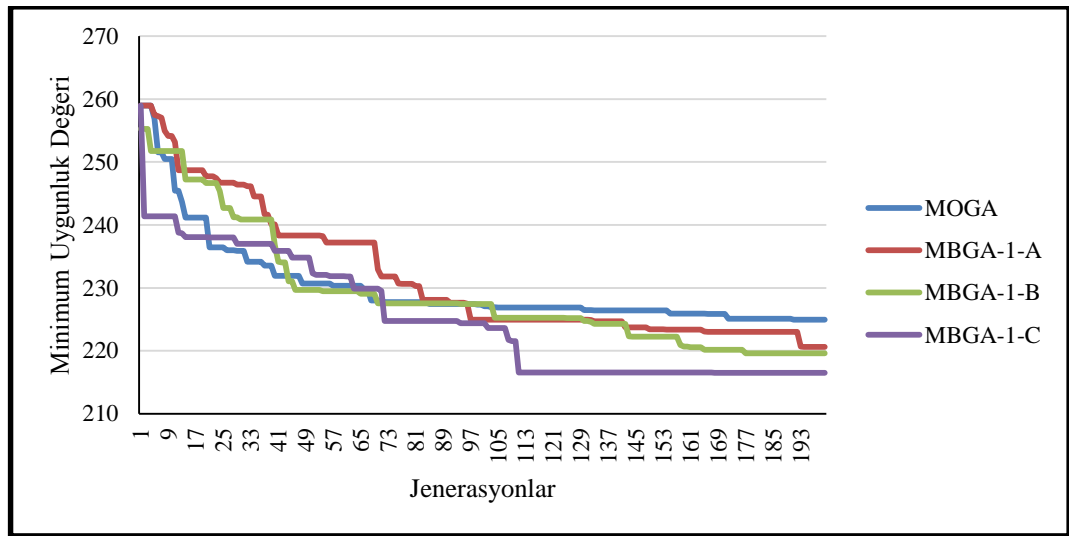
Şekil 6.4. 100 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

15 iş test problemi $f(x)_{min}$ değerlerine ait Şekil 6.5.'teki grafik incelendiğinde, ilk jenerasyonlarda MBGA-1-C'nin en düşük $f(x)_{min}$ değerini sağladığı tespit edilmiştir. Jenerasyon sayısındaki artış ile birlikte MBGA-1-B algoritması diğer 3 algoritma ile kıyaslandığında en küçük $f(x)_{min}$ değerinin elde edilmesini sağlamıştır. Son jenerasyonlar incelendiğinde MBGA-1 algoritmaları ile MOGA'dan daha düşük $f(x)_{min}$ değerleri sağlandığı görülmektedir.



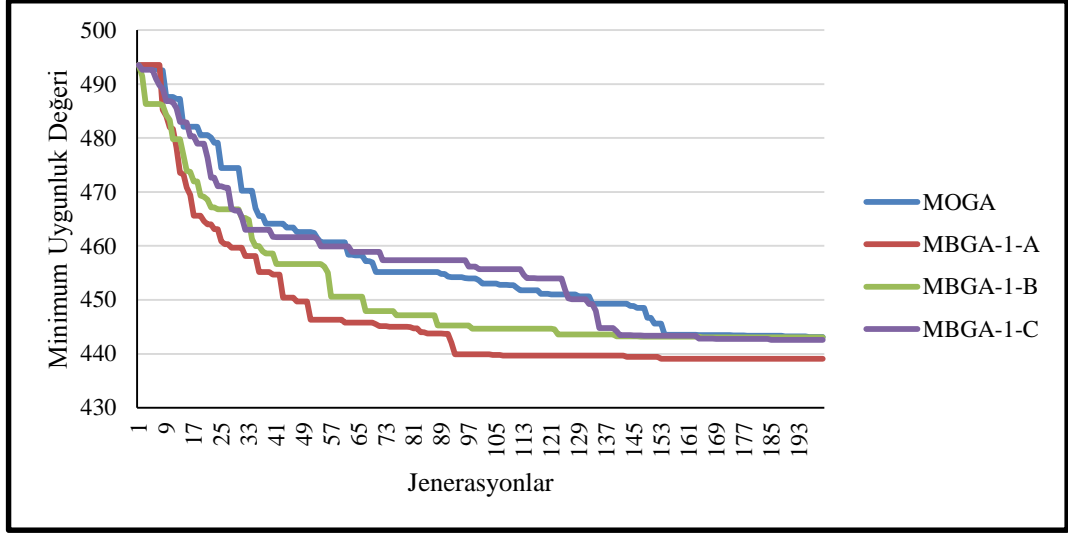
Şekil 6.5. 15 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

25 iş problemine ait sonuçların yer aldığı Şekil 6.6'daki grafik incelendiğinde MBGA-1-C'nin ilk jenerasyonlarda kıyaslanan diğer algoritmalarından daha düşük $f(x)_{min}$ değerine ulaştığı görülmektedir. Jenerasyonların ilerlemesi ile MOGA ve MBGA-1-B algoritmalarının performansında artış meydana gelmiştir. Son jenerasyonlar incelendiğinde, MBGA-1-C algoritması ile en düşük $f(x)_{min}$ değerine ulaşıldığı ve algoritmanın MOGA'dan daha başarılı olduğu tespit edilmiştir. MBGA-1-A ve MBGA-1-B algoritmaları da son jenerasyonlarda MOGA'dan daha düşük $f(x)_{min}$ değerlerine ulaşmıştır.



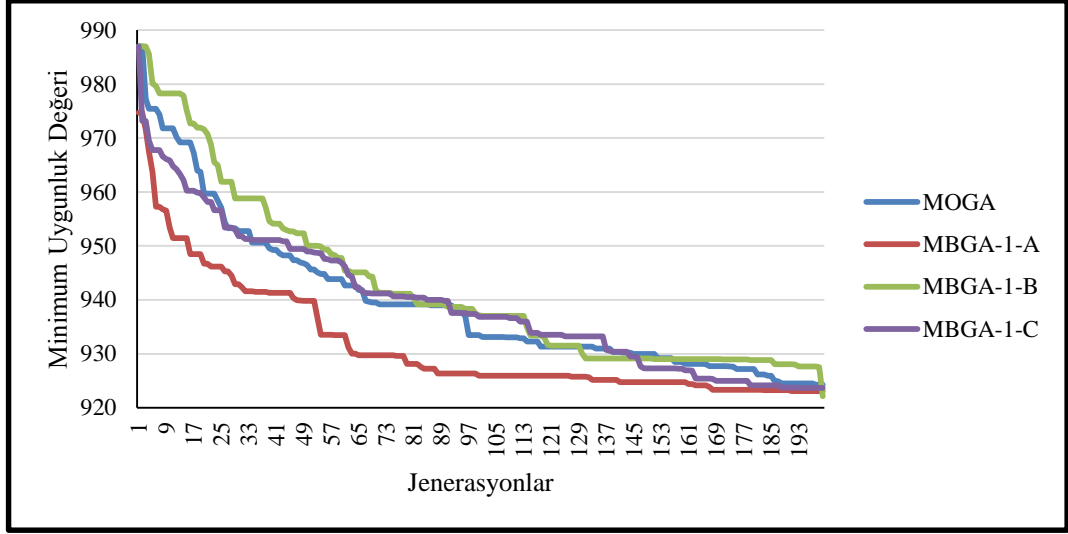
Şekil 6.6. 25 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Minimum uygunluk değerlerinin 50 işten oluşan test problemi üzerindeki değişimine ait Şekil 6.7.'deki grafik incelendiğinde, MBGA-1-A algoritmasının diğer algoritmalarla göre erken minimizasyon sağladığı ve jenerasyonlar sonunda en düşük $f(x)_{min}$ değerini elde ettiği görülmektedir. Erken minimizasyon sağlayan bir diğer algoritma da MBGA-1-B algoritması olmuştur. Son jenerasyonlarda MBGA-1-B, MBGA-1-C ve MOGA algoritmalarının birbirlerine yakın $f(x)_{min}$ değerleri sağladığı tespit edilmiştir.



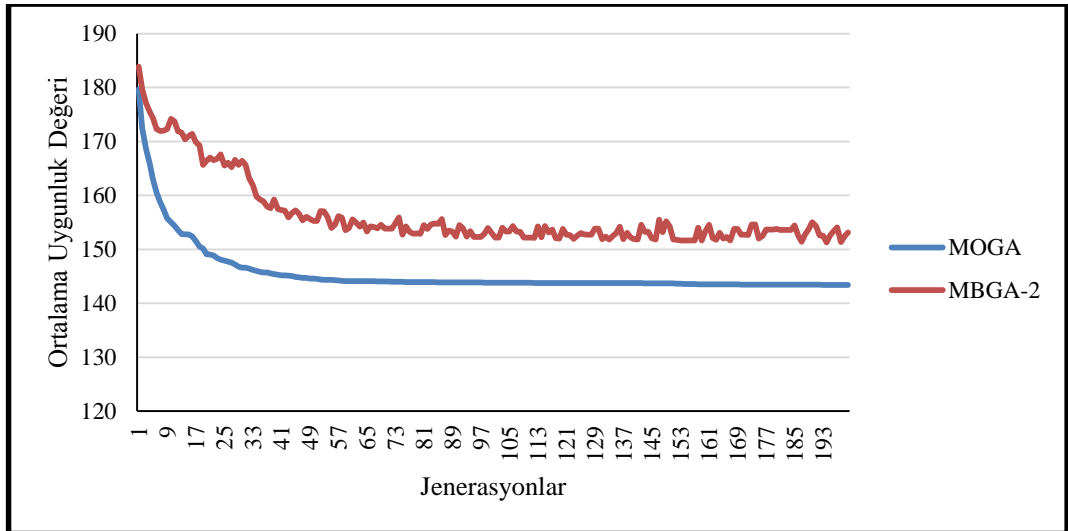
Şekil 6.7. 50 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.8.'de yer alan grafik incelendiğinde, 100 iş test probleminde MBGA-1-A algoritmasının daha kısa zamanda $f(x)_{min}$ değerinde azalma sağladığı görülmektedir. İlk jenerasyonlarda düşük $f(x)_{min}$ değerleri sağlayan MBGA-1-C algoritması, jenerasyon sayısının artışı ile MBGA-1-B ve MOGA algoritmaları ile yakın $f(x)_{min}$ değerlerine ulaşmıştır. Jenerasyonlar boyunca MBGA-1-B ve MBGA-1-C algoritmaları MOGA'ya yakın bir seyir izlemiş ve bu algoritmalar ile birbirlerine oldukça yakın sonuçlar elde edilmiştir. Son jenerasyonlar incelendiğinde MBGA-1 algoritmaları ile MOGA'ya oldukça yakın fakat daha düşük $f(x)_{min}$ değerleri sağlandığı tespit edilmiştir.



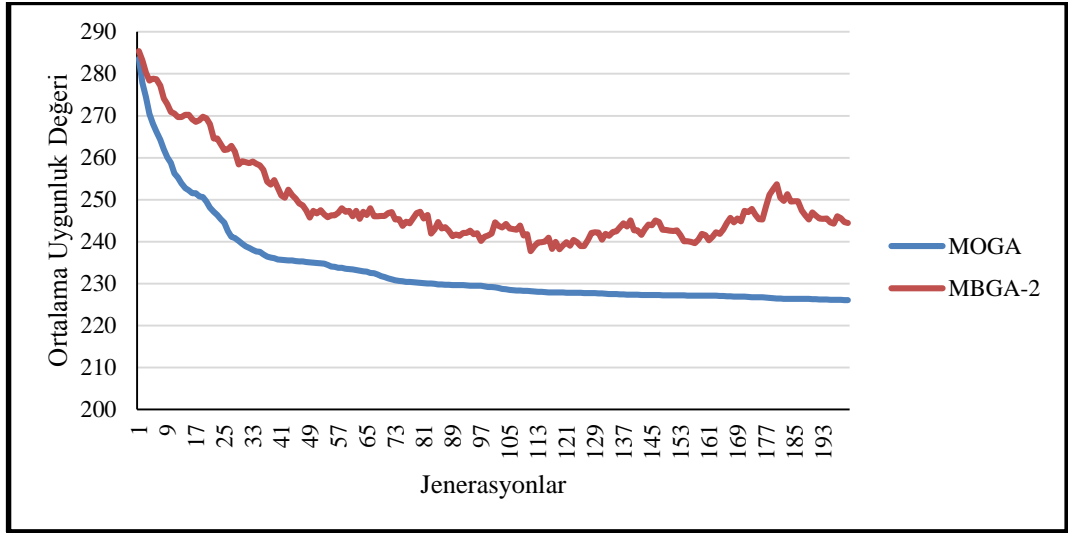
Şekil 6.8. 100 iş test problemi MBGA-1 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.9.'da yer alan 15 iş test problemi sonuçları incelendiğinde, $f(x)_{ort}$ değerleri açısından jenerasyonlar boyunca MOGA'nın MBGA-2 algoritması ile kıyaslandığında daha yüksek performans gösterdiği görülmektedir. MOGA, ilk jenerasyonlardan son jenerasyonlara kadar, $f(x)_{ort}$ değerlerinde daha fazla minimizasyon sağlamıştır.



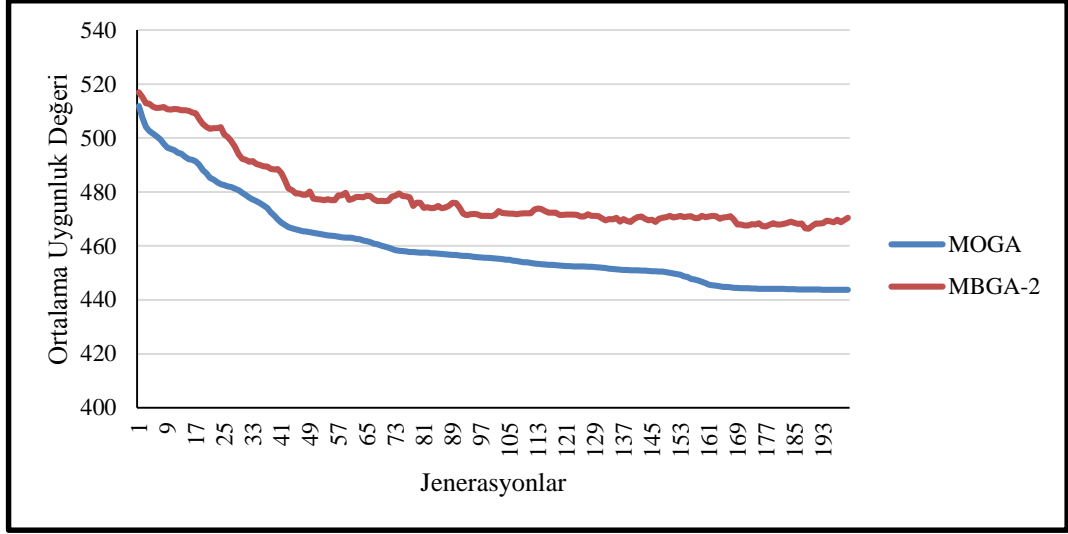
Şekil 6.9. 15 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

25 iş test problemi sonuçlarının yer aldığı Şekil 6.10.'daki grafik değerlendirildiğinde, ilk jenerasyondan son jenerasyona kadar MBGA-2'nin grafik üzerinde MOGA'dan daha yukarıda bir eğri izlediği ve performansının MOGA'nın gerisinde kaldığı görülmektedir. MOGA ile MBGA-2'ye oranla daha düşük $f(x)_{ort}$ değerleri elde edilmiştir.



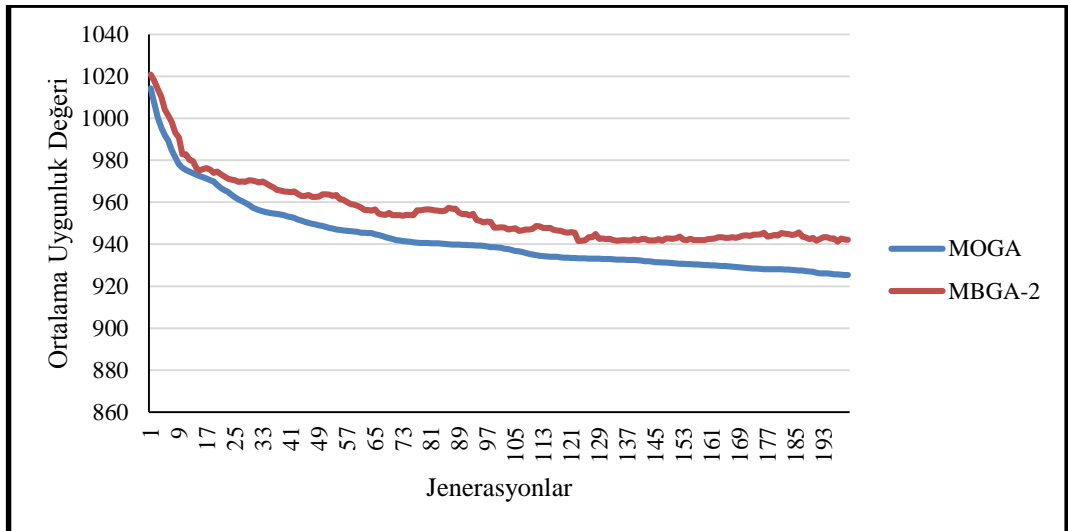
Şekil 6.10. 25 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

Şekil 6.11.'deki grafik ile gösterilen 50 iş test problemine ait sonuçlar, MOGA'nın MBGA-2 ile kıyaslandığında $f(x)_{ort}$ değerlerinin minimizasyonunda daha başarılı olduğunu göstermektedir.



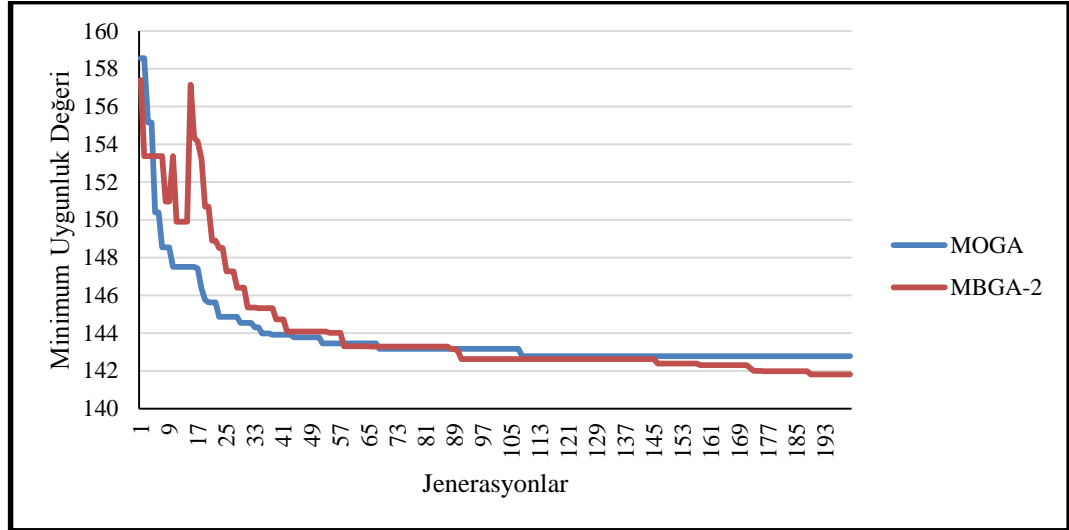
Şekil 6.11. 50 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

100 işten oluşan test problemi sonuçlarına ait Şekil 6.12.'de yer alan grafik incelendiğinde MOGA ve MBGA-2 algoritmalarının grafikte ilk jenerasyonlarda birbirine oldukça yakın $f(x)_{ort}$ değerleri sağladığı görülmektedir. MOGA, MBGA-2 algoritmasına yakın olmakla birlikte grafikte daha aşağıda bir eğriye sahiptir ve jenerasyonların artışı ile MBGA-2'ye kıyasla $f(x)_{ort}$ değerlerinde daha yüksek oranda azalma sağlamıştır.



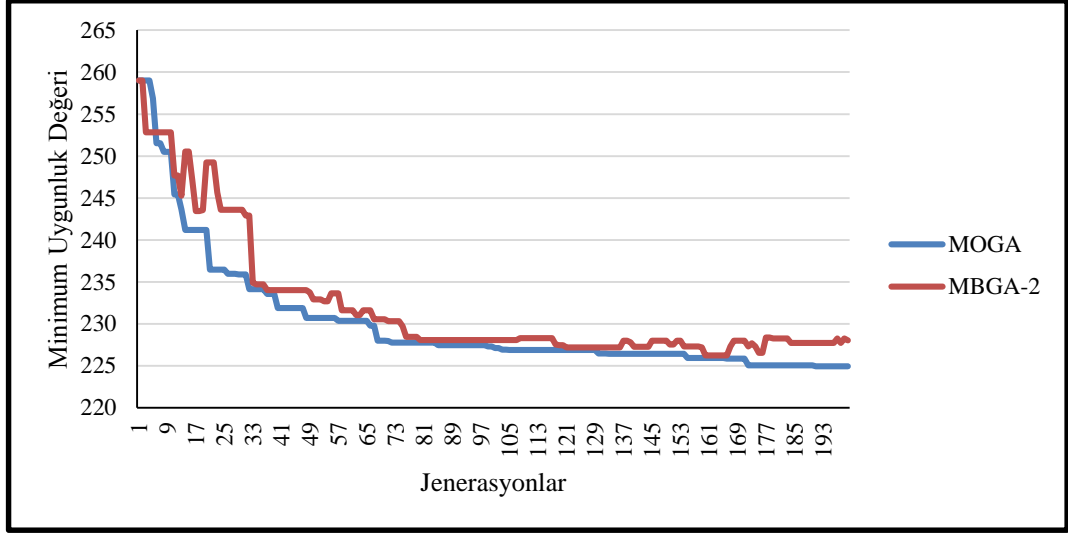
Şekil 6.12. 100 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

15 iş test problemi sonuçlarının yer aldığı Şekil 6.13.'teki grafik incelendiğinde ilk jenerasyonlarda MBGA-2 algoritmasının daha düşük $f(x)_{min}$ değerine ulaştığı görülmektedir. Jenerasyon sayısının artışı ile MOGA daha düşük $f(x)_{min}$ değerlerine ulaşmıştır. Son jenerasyonlar incelendiğinde en düşük $f(x)_{min}$ değerinin MBGA-2 algoritması ile elde edildiği tespit edilmiştir.



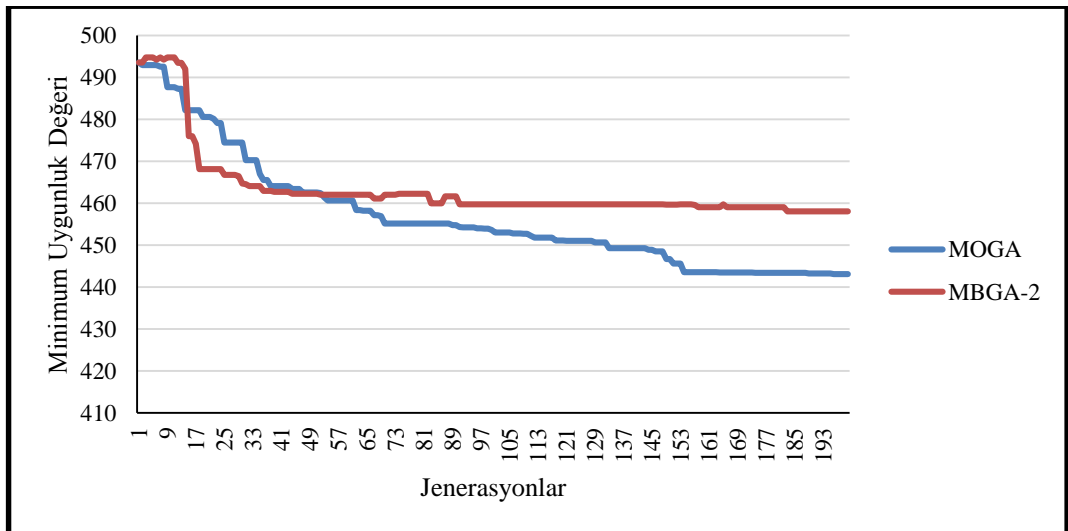
Şekil 6.13. 15 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.14.'teki grafik incelendiğinde, ilk jenerasyonlarda MBGA-2 algoritmasının $f(x)_{min}$ değerinde daha erken azalma sağladığı, jenerasyonlar ortasında MOGA ile daha düşük $f(x)_{min}$ değerleri elde edildiği görülmektedir. Jenerasyonlarda MOGA ile yakın sonuçlar sağlayan MBGA-2 algoritmasının son jenerasyonlara doğru oluşturduğu $f(x)_{min}$ değerleri yükselmiş ve uygulama sonucu MOGA'nın performansının daha yüksek olduğu tespit edilmiştir.



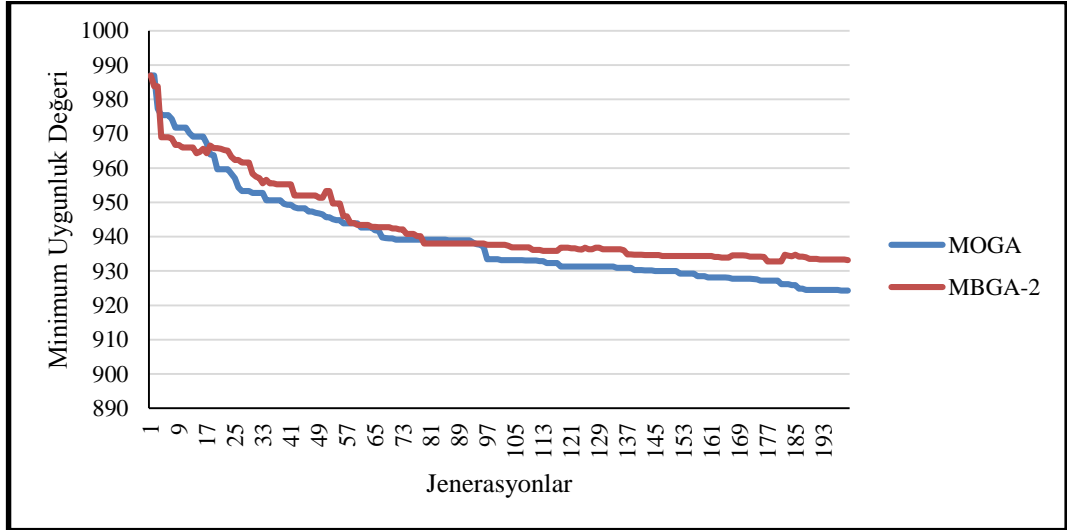
Şekil 6.14. 25 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

50 iş test problemi üzerinde gerçekleştirilen uygulamaya ait sonuçların Şekil 6.15.'te gösterilen grafiği incelendiğinde, jenerasyonlar ortasına kadar MBGA-2 ve MOGA algoritmalarının birbirlerine üstünlüklerinin değişkenlik gösterdiği, ilerleyen jenerasyonlarda MOGA'nın daha yüksek bir performansa sahip olduğu görülmektedir. Son jenerasyonlar incelendiğinde en küçük $f(x)_{min}$ değerinin MOGA ile elde edildiği tespit edilmiştir.



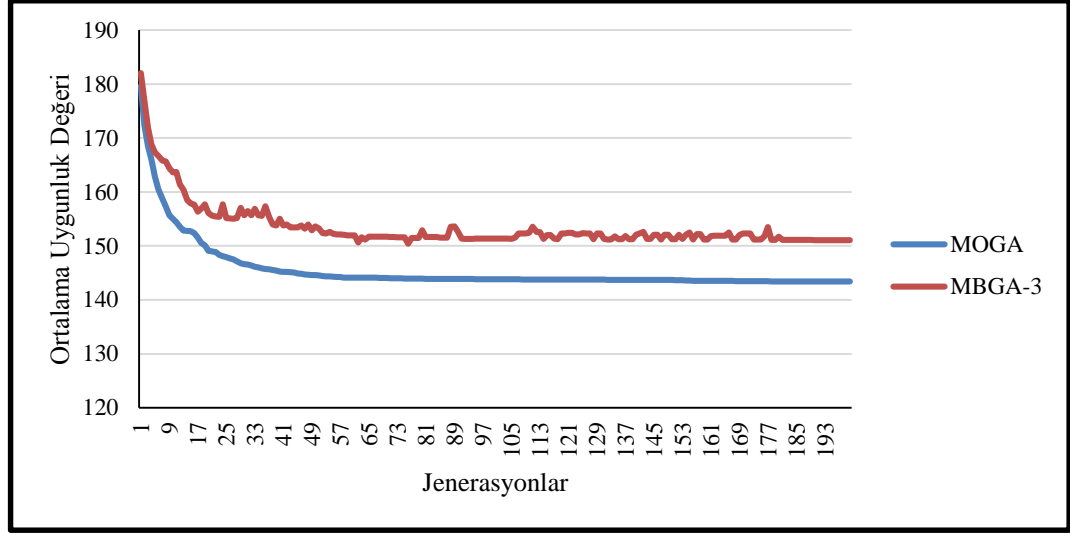
Şekil 6.15. 50 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.16.'da yer alan 100 iş test problemi sonuçları, ilk jenerasyonlarda MBGA-2'nin MOGA'dan daha düşük $f(x)_{min}$ değeri elde ettiğini göstermektedir. Jenerasyonların ilerlemesi ile MOGA'nın performansı artmış ve MBGA-2 ile yakın $f(x)_{min}$ değerleri oluşturmuştur. Son jenerasyonlarda ise MOGA, $f(x)_{min}$ değerlerinde daha fazla azalma sağlayarak en küçük $f(x)_{min}$ değerini elde etmiştir.



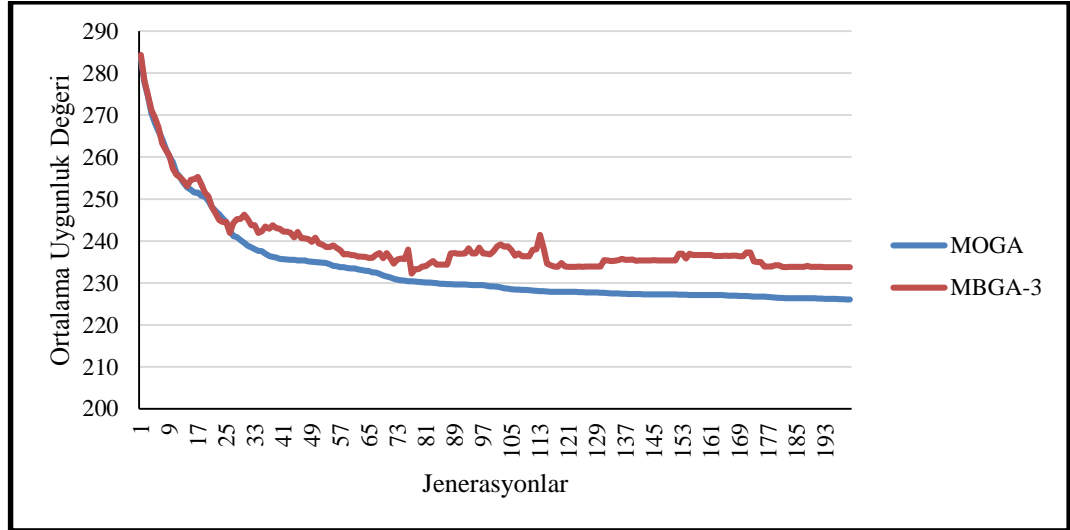
Şekil 6.16. 100 iş test problemi MBGA-2 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.17.'deki sonuçlar değerlendirildiğinde, ilk jenerasyonlarda $f(x)_{ort}$ değerlerinde yakın düzeylerde azalma sağlayan MBGA-3 ve MOGA algoritmalarının, jenerasyonların ilerlemesi ile grafik üzerinde eğrilerinin birbirlerinden uzaklaştığı görülmektedir. MOGA, 15 iş test problemi üzerinde $f(x)_{ort}$ değerlerinde MBGA-3'e göre daha fazla azalma sağlamış ve daha yüksek performans göstermiştir.



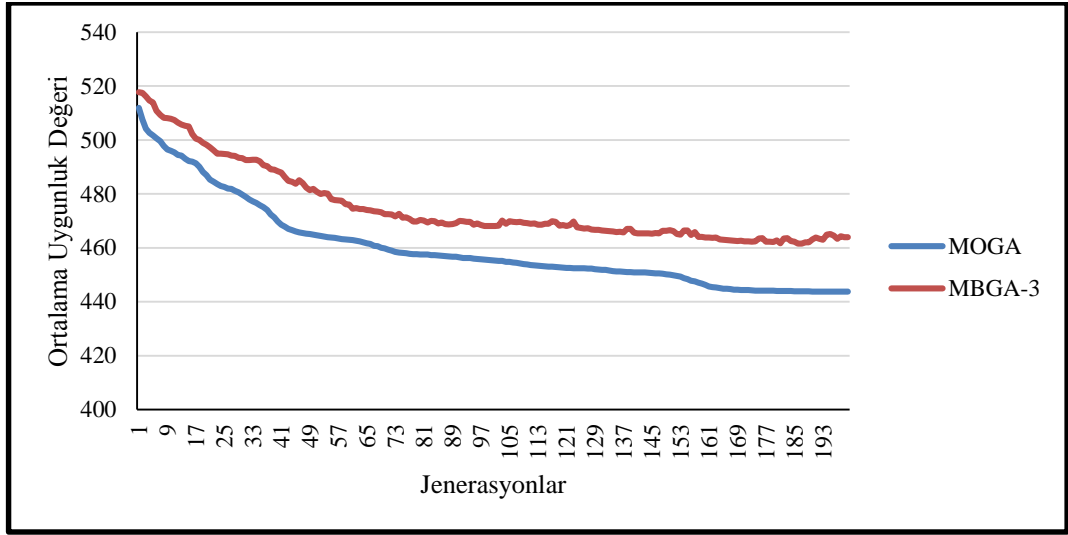
Şekil 6.17. 15 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

25 iş sonuçlarının gösterildiği Şekil 6.18.'de yer alan grafik incelendiğinde, ilk jenerasyonlarda grafik üzerinde birbirine yakın eğrilere sahip olan MBGA-3 ve MOGA'nın, ilerleyen jenerasyonlarda eğrilerinin birbirlerinden uzaklaştığı görülmektedir. Jenerasyonlar sonunda $f(x)_{ort}$ değerlerinde MOGA'nın daha fazla azalma sağladığı tespit edilmiştir.



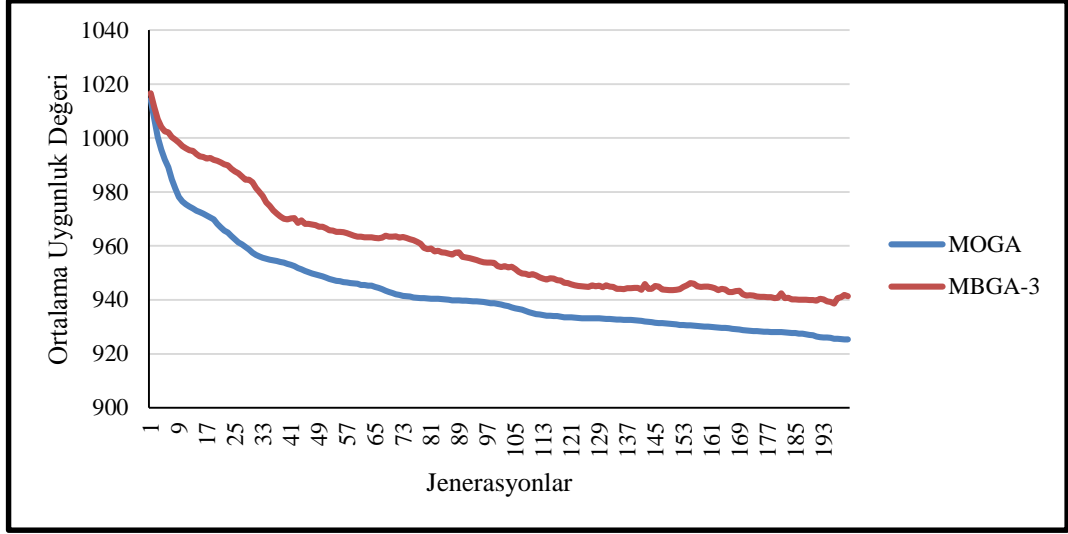
Şekil 6.18. 25 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

50 iş sonuçlarının grafiksel olarak gösterildiği Şekil 6.19.'da, ilk jenerasyonlardan son jenerasyonlara kadar $f(x)_{ort}$ değerlerinde MOGA'nın, MBGA-3 algoritmasından daha fazla azalma sağladığı görülmektedir.



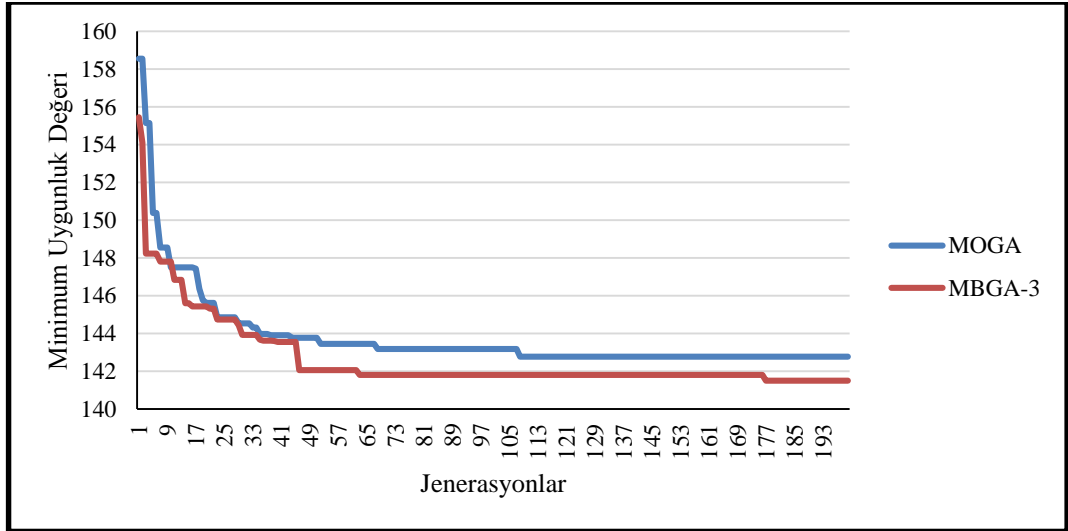
Şekil 6.19. 50 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

MBGA-3 ve MOGA algoritmalarının $f(x)_{ort}$ değerinde meydana gelen değişim açısından 100 iş test problemi üzerindeki performansları, Şekil 6.20.'de yer alan grafik üzerinde gösterilmiştir. Grafik incelendiğinde, ilk jenerasyonlarda yakın performans göstermelerine karşın, MOGA'nın performansını arttırdığı ve $f(x)_{ort}$ değerlerinde daha fazla azalma sağlayarak, daha başarılı olduğu görülmektedir.



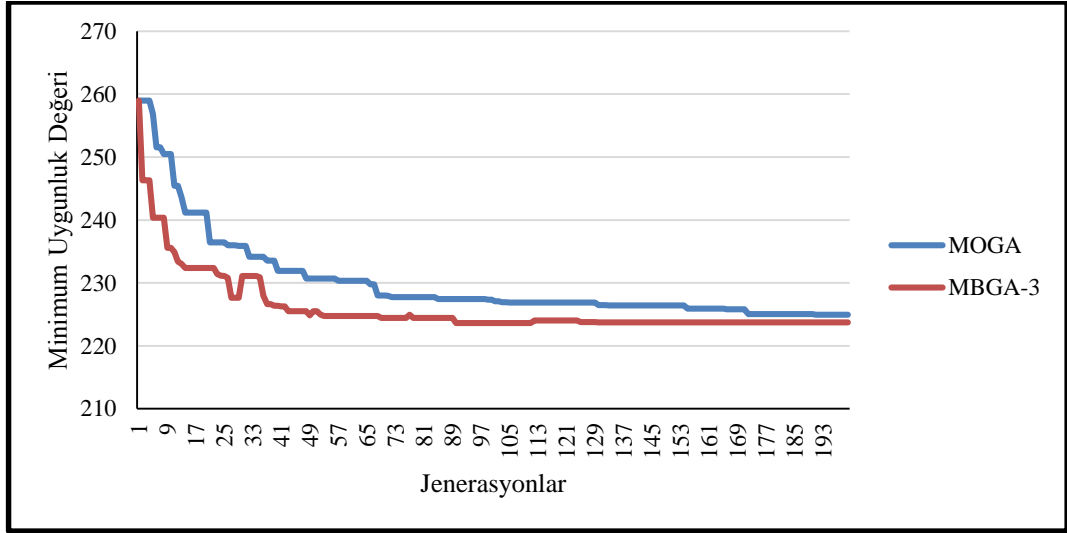
Şekil 6.20. 100 iş test problemi MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

Şekil 6.21'de yer alan 15 iş sonuçlarına ait grafik incelendiğinde, yaklaşık olarak ilk 50 jenerasyonda MBGA-3'ün MOGA'dan daha iyi $f(x)_{min}$ değerleri elde ettiği görülmektedir. İlerleyen jenerasyonlarda algoritmalar oldukça yakın sonuçlar sağlamış, son jenerasyonlara doğru MBGA-3 algoritması daha başarılı olmuştur. Jenerasyonlar tamamlandığında en küçük $f(x)_{min}$ değeri MBGA-3 ile elde edilmiştir.



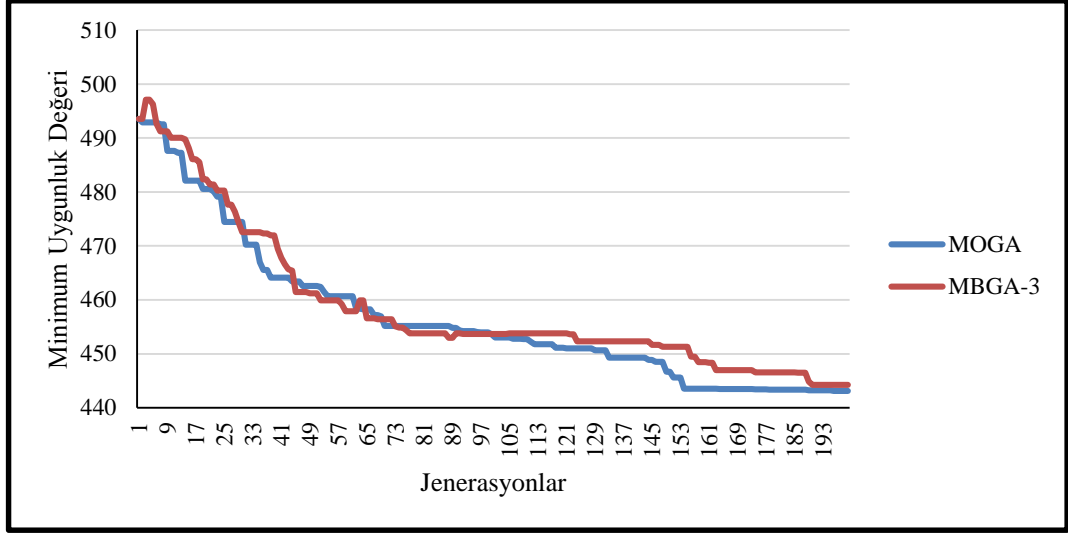
Şekil 6.21. 15 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

25 iş sonuçlarına ait Şekil 6.22.'de yer alan grafik incelendiğinde jenerasyonlar boyunca MBGA-3 algoritmasının $f(x)_{min}$ değerleri açısından MOGA'ya üstünlük sağladığı görülmektedir. Jenerasyonlar sonunda algoritmalara ait sonuçlar yakın değerlere ulaşmış olsa da, en küçük $f(x)_{min}$ değeri MBGA-3 ile elde edilmiştir.



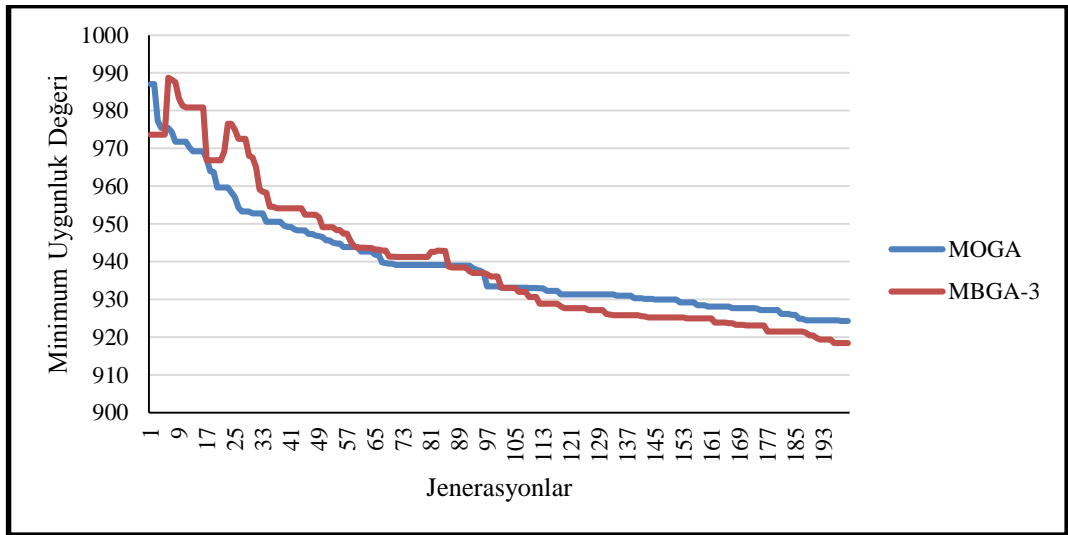
Şekil 6.22. 25 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.23.'te yer alan grafik sonuçlarına göre, MOGA ve MBGA-3 algoritmalarına ait eğrilerin jenerasyonlar boyunca birbirlerine oldukça yakın olduğu ve algoritmaların benzer sonuç sağladığı tespit edilmiştir. Son jenerasyonlar incelendiğinde, MOGA ile en küçük $f(x)_{min}$ değerlerine ulaşıldığı görülmektedir.



Şekil 6.23. 50 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

100 iş sonuçlarının yer aldığı Şekil 6.24'teki grafik incelendiğinde, MOGA ve MBGA-3 algoritmalarının birbirlerine olan üstünlüklerinin değişkenlik gösterdiği, jenerasyon sayısının artışı ile MBGA-3 algoritmasının MOGA'dan daha başarılı olduğu ve en küçük $f(x)_{min}$ değerinin bu algoritma ile elde edildiği görülmüştür.

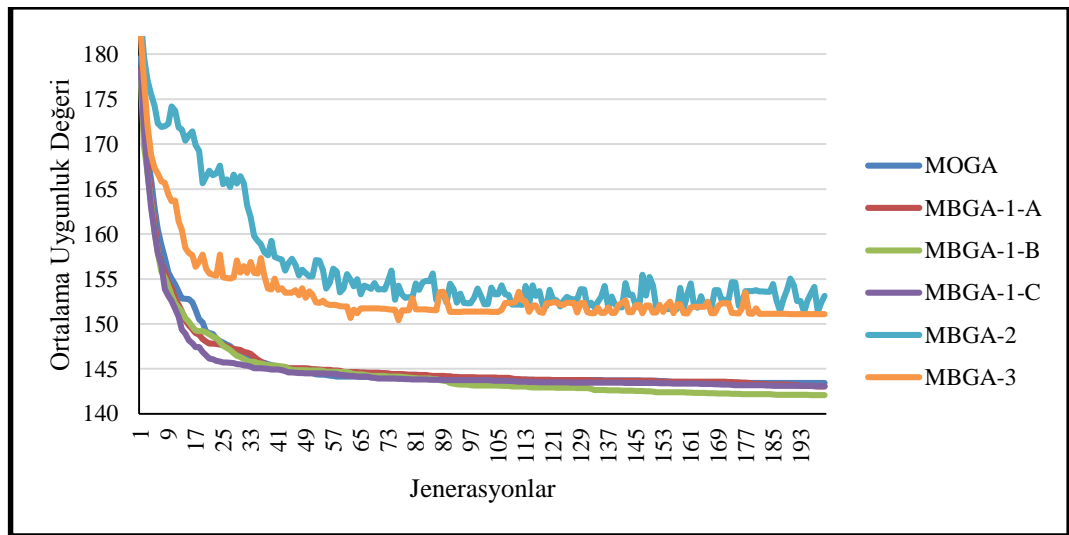


Şekil 6.24. 100 iş test problemi MBGA-3 algoritması ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Test problemleri üzerinde, geliştirilen MBGA-1, MBGA-2 ve MBGA-3 algoritmaları ile yapılan uygulamalar sonucu algoritma performanslarının değerlendirilmesinde $f(x)_{ort}$ ve $f(x)_{min}$ değerlerine ait elde edilen sonuçlar doğrultusunda, 5 hibrit yapının

birlikte yer aldığı grafikler oluşturulmuştur. Algoritmaların 200 jenerasyon boyunca $f(x)$ değerlerinde meydana getirdiği değişikliklerden yararlanılarak, algoritmalar birbirleri ve MOGA ile kıyaslanmıştır.

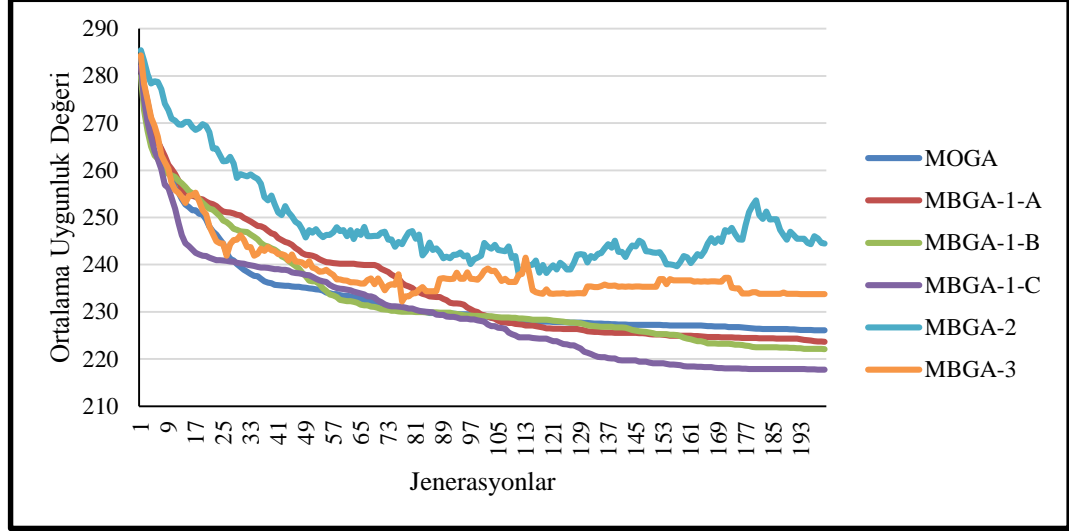
Şekil 6.25'te yer alan 15 işten oluşan test problemine ait $f(x)_{ort}$ değerleri grafiği incelendiğinde MBGA-2 ve MBGA-3 hibrit algoritmalarının en yüksek $f(x)_{ort}$ değerlerini elde ettiği görülmektedir. MBGA-1-A, MBGA-1-B, MBGA-1-C algoritmaları MOGA ile grafik üzerinde oldukça yakın eğriler oluşturmuştur. Minimum değerlere ulaşma süresi bakımından MBGA-1-C algoritması $f(x)_{ort}$ değerlerinde daha kısa sürede azalma sağlamıştır. Son jenerasyonlar incelendiğinde MBGA-1-B'nin MOGA'ya üstünlük sağladığı ve popülasyon kalitesini daha fazla arttırdığı tespit edilmiştir.



Şekil 6.25. 15 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

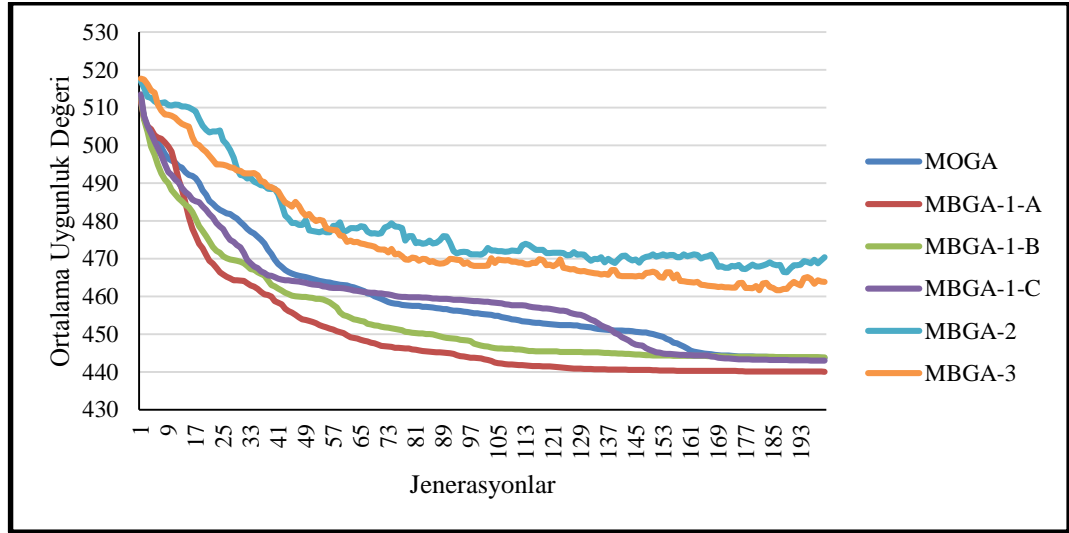
25 iş test problemi sonuçlarının yer aldığı Şekil 6.26'daki grafik incelendiğinde, MBGA-1-C algoritmasının, $f(x)_{ort}$ değerlerinde diğer algoritmalarla oranla ilk jenerasyonlarda daha fazla azalma sağladığı görülmektedir. MBGA-2 algoritması, jenerasyonlar boyunca diğer algoritmaların gerisinde performans göstermiştir. İlk jenerasyonlarda $f(x)_{ort}$ değerlerinde erken azalma sağlayarak başarılı olan MBGA-3 algoritması jenerasyon sayısının artması ile diğer algoritmalar kadar azalma sağlayamayarak düşük bir performans göstermiştir. $f(x)_{ort}$ değerlerinde en büyük

azalma oranını sağlayan MBGA-1-C algoritması, bu test problemi üzerinde başarısı en yüksek olan algoritma olmuştur.



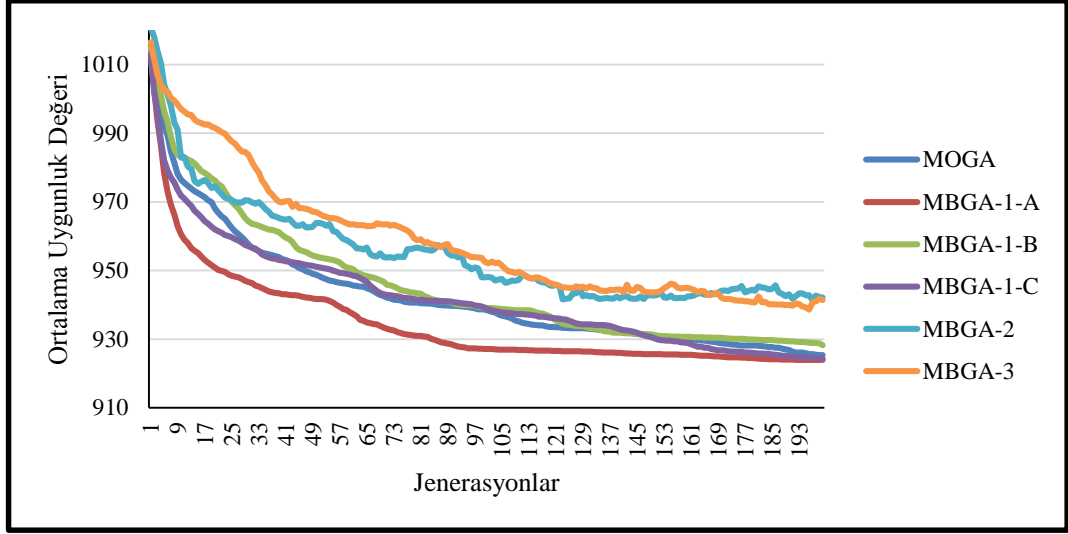
Şekil 6.26. 25 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

50 iş test problemi üzerinde yapılan uygulama ile Şekil 6.27.'de yer alan grafik sonuçları elde edilmiştir. Sonuçlar incelendiğinde, MBGA-2 ve MBGA-3 algoritmalarının jenerasyonlar boyunca $f(x)_{ort}$ değerlerinin yüksek olduğu görülmektedir. MBGA-1-B ve MBGA-1-C algoritmaları, ilk jenerasyonlarda MOGA'ya oranla $f(x)_{ort}$ değerlerinde daha fazla azalma sağlarken, jenerasyon sayısının artışı ile birlikte MOGA ile grafik üzerinde oldukça yakın eğriler oluşturmuş ve yakın bir başarı sağlamıştır. Bu test problemi üzerinde yapılan uygulamada en iyi performans, MBGA-1-A algoritması ile sağlanmıştır. MBGA-1-A, MOGA'ya ve diğer algoritmalarla kıyasla $f(x)_{ort}$ değerlerinde daha yüksek oranda azalma sağlamış ve oldukça başarılı bir performans göstermiştir.



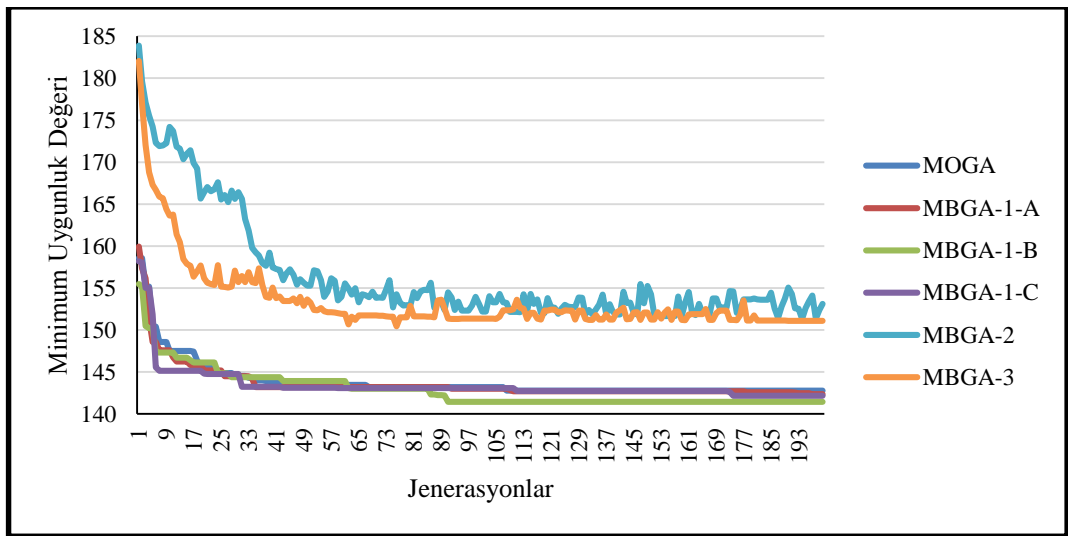
Şekil 6.27. 50 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

Şekil 6.28.'deki grafik üzerinde gösterilmiş olan 100 iş test problemine ait sonuçlar incelendiğinde, ilk jenerasyonlarda MBGA-1-A ve MBGA-1-C algoritmalarının MOGA ve diğer algoritmalarla göre, daha önceki jenerasyonlarda $f(x)_{ort}$ değerlerinde azalma sağladığı görülmektedir. Başlangıçta MOGA ile yakın bir performans gösteren MBGA-1-B algoritması, jenerasyonlar sonuna dek performansını sürdürmüştür. MBGA-2 ve MBGA-3 algoritmaları diğer algoritmalarla kıyasla daha yüksek $f(x)_{ort}$ değerleri elde etmiştir. Son jenerasyonlar incelendiğinde, $f(x)_{ort}$ değerlerinde MBGA-1-A algoritması ile en yüksek oranda azalma sağlandığı tespit edilmiştir. Jenerasyonlar sonunda MBGA-1-B, MBGA-1-C ve MOGA ile birbirlerine yakın $f(x)_{ort}$ değerleri elde edilirken, MBGA-2 ve MBGA-3 algoritmalarının performansının düşük olduğu görülmektedir.



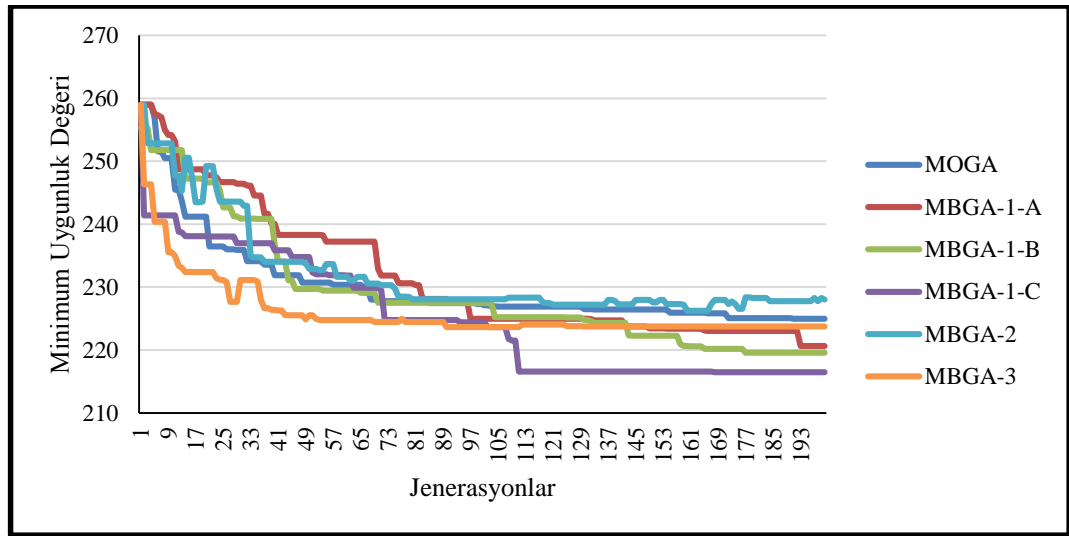
Şekil 6.28. 100 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{ort}$ değerleri değişimi grafiği

15 iş test problemi sonuçlarının yer aldığı Şekil 6.29.'da yer alan grafik incelendiğinde MBGA-2 ve MBGA-3 algoritmalarının MOGA'dan ve diğer hibrit algoritmalarından daha düşük performans gösterdiği görülmektedir. İlk jenerasyonlarda en düşük $f(x)_{min}$ değerini sağlayan MBGA-1-A algoritması, MBGA-1-C ve MOGA ile $f(x)_{min}$ değerleri açısından ilerleyen jenerasyonlarda oldukça yakın performans göstermiştir. Son jenerasyonlar incelendiğinde en küçük $f(x)_{min}$ değerlerine MBGA-1-B algoritması ile ulaşıldığı tespit edilmiştir.



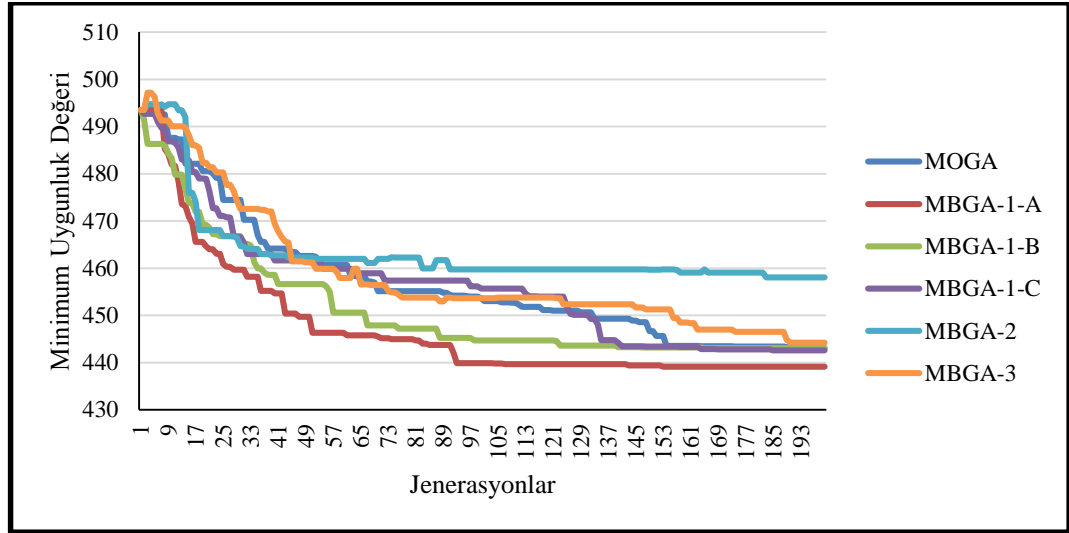
Şekil 6.29. 15 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Şekil 6.30.'da yer alan 25 iş test problemi sonuçları incelendiğinde, ilk jenerasyonlarda en düşük $f(x)_{min}$ değerlerinin MBGA-3 ile elde edildiği görülmektedir. İlk jenerasyonlarda MBGA-3 kadar başarılı $f(x)_{min}$ değerleri sağlayamasa da MBGA-1-C algoritması, MBGA-1-A, MBGA-1-B, MBGA-2 ve MOGA'dan daha düşük $f(x)_{min}$ değerlerine ulaşmıştır. Jenerasyon sayısının artışı ile algoritmalara ait $f(x)_{min}$ değerleri birbirine yaklaşmış, jenerasyon sayısının ortalarından itibaren MBGA-1-C algoritması en küçük $f(x)_{min}$ değerini elde ederek, en yüksek performansı göstermiştir.



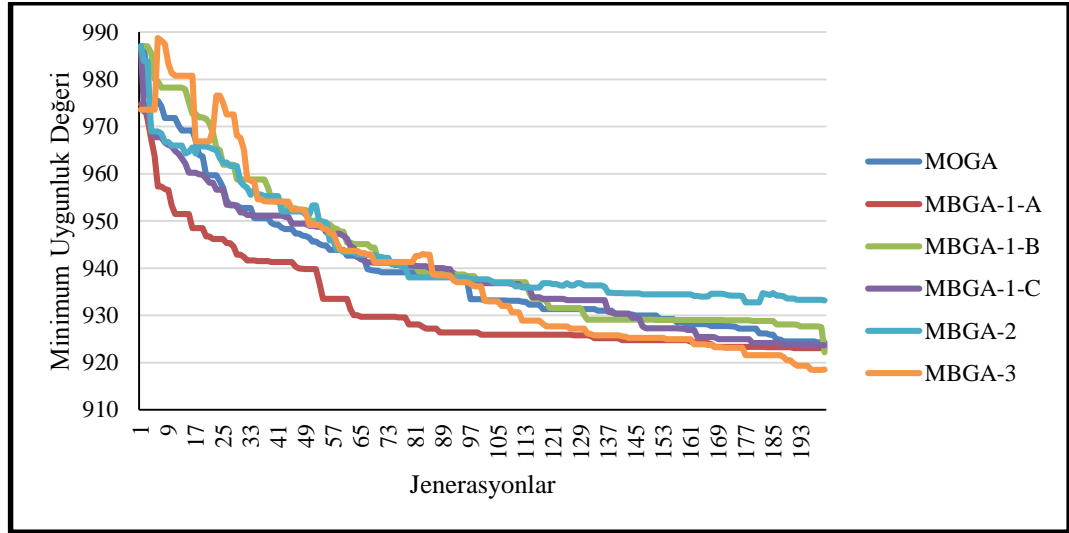
Şekil 6.30. 25 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Algoritmaların 50 iş test problemi üzerindeki sonuçlarının yer aldığı Şekil 6.31.'deki grafik incelendiğinde, jenerasyonların başlangıcında MBGA-1-A algoritmasının kısa sürede en düşük $f(x)_{min}$ değerlerini elde ettiği ve jenerasyonlar sonunda da en yüksek başarıyı sağlayarak, en küçük $f(x)_{min}$ değerine ulaştığı görülmektedir. İlk jenerasyonlarda diğer algoritmalara yakın $f(x)_{min}$ değerleri elde eden MBGA-2 algoritması, jenerasyon sayısının artışı ile diğer algoritmalara oranla daha yüksek $f(x)_{min}$ değerleri elde ederek düşük bir performans göstermiştir. Son jenerasyonlara kadar MOGA ve MBGA-1-C'den daha başarılı sonuçlar elde eden MBGA-1-B algoritması, son jenerasyonlarda bu iki algoritma ile oldukça yakın değerlere ulaşmıştır.



Şekil 6.31. 50 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

100 iş test problemi sonuçlarının yer aldığı Şekil 6.32.'de gösterilen grafik incelendiğinde, ilk jenerasyonlarda MBGA-1-A'nın en yüksek performansı gösterdiği ve diğer algoritmaların birbirlerine yakın performanslar gösterdiği tespit edilmiştir. Son jenerasyonlar incelendiğinde, başlangıçta performansı düşük olan MBGA-3 algoritmasının en küçük $f(x)_{min}$ değerini elde ettiği görülmektedir. Jenerasyonlar sonunda MBGA-2 ile en yüksek $f(x)_{min}$ değerleri elde edilirken, MBGA-1-A, MBGA-1-B, MBGA-1-C ve MOGA ile birbirlerine oldukça yakın $f(x)_{min}$ değerleri elde edilmiştir.



Şekil 6.32. 100 iş test problemi MBGA-1, MBGA-2, MBGA-3 algoritmaları ve MOGA'nın 200 jenerasyon $f(x)_{min}$ değerleri değişimi grafiği

Genetik algoritmaların seçim adımına MultiMoora'nın 3 farklı şekilde entegre edilmesi ile oluşturulan MBGA-1 olarak gruplandırılan MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmaları 4 test problemi üzerinde uygulanmıştır. 15, 25, 50 ve 100 iş sayılarından oluşan bu test problemleri üzerinde gerçekleştirilen uygulama sonucu her algoritmanın sağladığı $f(x)_{min}$ değerleri ve bu değerlerin elde edilmesi amacıyla algoritmaların optimize ettiği maksimum tamamlanma zamanı C_{max} , ortalama akış süresi \bar{F} , maksimum gecikme T_{max} , ortalama gecikme \bar{T} ve geciken iş sayısı NLJ kriterlerine ait değerler, MOGA'ya ait sonuçlar ile Tablo 6.1., Tablo 6.2., Tablo 6.3. ve Tablo 6.4. üzerinde karşılaştırılmıştır.

Algoritmaların 15 iş test problemi üzerinde çalıştırılması sonucu elde edilen ve Tablo 6.1.'de yer alan $f(x)_{min}$ değerleri kıyaslandığında, MultiMoora metodunun genetik algoritmaların seçim adımına entegre edilmesi ile oluşturulan MBGA-1-B'nin, en iyi $f(x)_{min}$ değerini (141,44) sağladığı görülmektedir. Geliştirilen MBGA-1-A, MBGA-1-C, MBGA-2 ve MBGA-3 algoritmalarının performansının MBGA-1-B algoritmasının gerisinde kaldığı fakat klasik çok amaçlı genetik algoritmayı ifade eden MOGA'dan daha yüksek olduğu tespit edilmiştir.

15 İŞ						
	$f(x)$	C_{max}	\bar{F}	NLJ	\bar{T}	T_{max}
MOGA	142,7733	429	284,8667	0	0	0
MBGA-1-A	142,3466667	429	282,7333333	0	0	0
MBGA-1-B	141,44	429	278,2	0	0	0
MBGA-1-C	142,16	429	281,8	0	0	0
MBGA-2	141,8133333	430	279,0666667	0	0	0
MBGA-3	141,4933333	429	278,4666667	0	0	0

Tablo 6.1. 15 iş test problemi üzerinde algoritmaların $f(x)_{min}$ değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu

25 işten oluşan test problemine ait Tablo 6.2.'de gösterilen sonuçlar incelendiğinde en iyi $f(x)_{min}$ değerinin, 216,48 uygunluk değeri ile MBGA-1-C hibrit algoritması tarafından sağlandığı görülmektedir. MBGA-2 ile en yüksek $f(x)_{min}$ değeri elde edilmiş ve algoritmanın performansı MOGA'nın gerisinde kalmıştır. Geliştirilen MBGA-1-A, MBGA-1-B ve MBGA-3 algoritmaları ile MBGA-1-C algoritması kadar minimizasyon sağlanamamış olmasa da, bu algoritmalar ile elde edilen sonuçlar MOGA'dan daha başarılı olmuştur.

25 İŞ						
	$f(x)$	C_{max}	\bar{F}	NLJ	\bar{T}	T_{max}
MOGA	224,944	594	372,84	11	27,88	119
MBGA-1-A	220,608	595	372,4	10	20,64	105
MBGA-1-B	219,584	595	372,36	7	18,56	105
MBGA-1-C	216,48	587	374,28	7	17,12	97
MBGA-2	228,032	600	373,72	9	32,44	125
MBGA-3	223,712	606	375,28	7	14,28	116

Tablo 6.2. 25 iş test problemi üzerinde algoritmaların $f(x)_{min}$ değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu

Geliştirilen 5 algoritmanın 50 iş test problemi sonuçlarının yer aldığı Tablo 6.3. incelendiğinde, MBGA-2 dışındaki tüm algoritmaların $f(x)_{min}$ değerinde MOGA'dan daha fazla minimizasyon sağladığı görülmektedir. 50 iş test probleminde

en başarılı sonuç ise 439,108 uygunluk değerinin elde edildiği MBGA-1-A hibrit algoritması ile sağlanmıştır.

50 İŞ						
	$f(x)$	C_{max}	\bar{F}	NLJ	\bar{T}	T_{max}
MOGA	443,116	974	551,54	32	174,04	484
MBGA-1-A	439,108	968	544,16	33	172,38	478
MBGA-1-B	443,056	972	560,64	31	179,64	472
MBGA-1-C	442,592	968	558,6	34	184,36	468
MBGA-2	458,052	982	564,42	33	188,84	522
MBGA-3	443,056	972	560,64	31	179,64	472

Tablo 6.3. 50 iş test problemi üzerinde algoritmaların $f(x)_{min}$ değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu

Tablo 6.4.'te yer alan 100 iş test problemi sonuçları değerlendirildiğinde, 918,48 uygunluk değerini sağlayan MBGA-3 algoritması ile diğer algoritmalarla kıyaslandığında daha fazla minimizasyon sağlandığı tespit edilmiştir. Diğer algoritmalarla ait sonuçlar incelendiğinde, MBGA-2 algoritması dışındaki tüm algoritmaların MOGA'dan daha yüksek performans gösterdiği görülmektedir.

100 İŞ						
	$f(x)$	C_{max}	\bar{F}	NLJ	\bar{T}	T_{max}
MOGA	924,298	1760	952,95	81	560,54	1267
MBGA-1-A	923,072	1767	950,27	82	559,09	1257
MBGA-1-B	922,16	1759	944	81	556,8	1270
MBGA-1-C	923,654	1770	957,39	82	568,88	1240
MBGA-2	933,158	1776	961,19	81	571,6	1276
MBGA-3	918,48	1757	946,48	80	557,92	1251

Tablo 6.4. 100 iş test problemi üzerinde algoritmaların $f(x)_{min}$ değerleri ve bu değerlerin hesaplandığı kriter değerleri tablosu

Grafiksel sonuçlar ve tablo sonuçları değerlendirildiğinde algoritmaların performanslarının $f(x)_{ort}$ ve $f(x)_{min}$ değerlerinin 15, 25, 50 ve 100 iş sayılarından

oluşan test problemleri üzerinde zaman zaman birbirlerine olan üstünlüklerinin değişkenlik gösterdiği görülmektedir.

Uygulama sonucu 200 jenerasyon boyunca elde edilen $f(x)_{ort}$ değerlerinin değişimlerinin yer aldığı grafikler, tüm iş sayıları göz önünde bulundurularak değerlendirildiğinde, MBGA-1-A algoritmasının MOGA'dan daha yüksek performans sağladığı görülmektedir. MBGA-1-A ile 15 iş test problemi $f(x)_{ort}$ değerlerinde MOGA ile yakın sonuç elde edilmiştir. 25, 50 ve 100 iş test problemleri üzerinde jenerasyonlarda elde edilen $f(x)_{ort}$ değerlerinde MBGA-1-A, MOGA'ya belirgin bir şekilde üstünlük sağlamıştır. MBGA-1-B algoritması, 15 ve 25 iş test problemlerinde MOGA'ya belirgin üstünlük sağlarken, 50 ve 100 iş test problemlerinde MOGA ile yakın performans göstermiştir. MBGA-1-C algoritmasının jenerasyonlar boyunca elde edilen $f(x)_{ort}$ değerleri incelendiğinde, 25 ve 100 iş test problemlerinde MOGA'dan daha başarılı olduğu görülmektedir. 15 ve 50 iş sayılarından oluşan test problemlerinde ise iki algoritmaya ait $f(x)_{ort}$ değerleri birbirlerine oldukça yakın düzeylere ulaşmıştır. Algoritmaların $f(x)_{ort}$ değerleri açısından değerlendirilmesi sonucu, MOGA'ya üstünlük sağlayan MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmalarından çok amaçlı akış tipi çizelgeleme problemlerinin çözümünde yararlanılabileceği görülmektedir.

Test problemleri üzerinde gerçekleştirilen uygulamalar sonucu jenerasyonlarda elde edilen $f(x)_{min}$ değerleri incelendiğinde, MultiMoora metodunun genetik algoritmaların seçim adımına entegre edilmesi ile oluşturulan ve MBGA-1 olarak gruplandırılan algoritmaların, MOGA'dan daha düşük $f(x)_{min}$ değerlerine ulaştığı tespit edilmiştir. MultiMoora metodunun seçim adımında değerlendirilmesi ile farklı varyasyonlar çerçevesinde geliştirilen MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmalarının 15, 25, 50 ve 100 iş test problemleri sonucunda $f(x)_{min}$ değerinde MOGA'dan daha fazla minimizasyon sağlaması, MultiMoora metodunun genetik algoritmaların seçim adımında çok amaçlı değerlendirmedeki performansını arttırdığını göstermektedir. Ayrıca MBGA-3 algoritması 15, 25, 50 ve 100 iş problemlerinde MOGA'ya üstünlük sağlamıştır. Bu nedenle $f(x)_{min}$ değerleri açısından değerlendirildiğinde, çok amaçlı akış tipi çizelgeleme problemlerine çözüm

getirmek amacıyla MBGA-1 grubu ve MBGA-3 algoritmalarından yararlanılabileceği sonucu elde edilmektedir.

Algoritmalara ait $f(x)_{ort}$ ve $f(x)_{min}$ değerlerinden oluşan sonuçlar birlikte değerlendirildiğinde, MultiMoora metodunun genetik algoritmaların seçim adımında yer aldığı, MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmalarının MOGA'ya üstünlük sağladığı görülmektedir. MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmaları ile çalışma sonunda $f(x)$ değerlerinde en başarılı sonuçlar elde edilmiştir. Algoritmaların MOGA'ya üstünlük sağlaması sebebiyle, çok amaçlı akış tipi çizelgeleme problemlerinde kullanımı önerilmektedir.

MultiMoora metodunun, MBGA-1-A, MBGA-1-B ve MBGA-1-C algoritmalarının başarıları gözlemlendiğinde, genetik algoritmaların seçim adımında başarılı bir, çok amaçlı değerlendirme sağladığı sonucuna ulaşılmaktadır. MultiMoora metodu ile seçim adımında elde edilen bu başarı, metodun genetik algoritmaların yalnızca eleme adımında ve hem seçim hem de eleme adımında birlikte yer aldığı durumlarda sağlanamamıştır. Çalışma ile çok amaçlı akış tipi çizelgeleme problemlerinde, genetik algoritmaların seçim adımına MultiMoora metodunun entegre edilmesi ile genetik algoritmaların uygulamada performansının arttırıldığı görülmektedir.

KAYNAKLAR

- Abd Rahman, R., Ramli, R., Jamari, Z., & Ku-Mahamud, K. R. (2016). Evolutionary Algorithm with Roulette-Tournament Selection for Solving Aquaculture Diet Formulation. *Mathematical Problems in Engineering*, 2016.
- Aydemir, E. (2009). Atölye Tipi Çizelgeleme Problemlerinin Öncelik Kuralı Tabanlı Genetik Algoritma Yaklaşımıyla Simülasyon Destekli Optimizasyonu. Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.
- Basseur, M., Seynhaeve, F., & Talbi, E. G. (2002). Design of multi-objective evolutionary algorithms: Application to the flow-shop scheduling problem. In *Congress on Evolutionary Computation CEC'02*, 2, 1151-1156.
- Blickle, T., & Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms.
- Braglia, M., & Grassi, A. (2009). A new heuristic for the flowshop scheduling problem to minimize makespan and maximum tardiness. *International Journal of Production Research*, 47(1), 273-288.
- Brauers, W. K. M., & Zavadskas, E. K. (2006). The MOORA method and its application to privatization in a transition economy. *Control and Cybernetics*, 35(2), 445-469.
- Brauers, W. K. M., & Zavadskas, E. K. (2010). Project management by MULTIMOORA as an instrument for transition economies. *Technological and Economic Development of Economy*, 16(1), 5-24.
- Brauers, W. K. M., & Zavadskas, E. K. (2012). Robustness of MULTIMOORA: a method for multi-objective optimization. *Informatica*, 23(1), 1-25.
- Chang, P. C., Chen, S. H., Fan, C. Y., & Chan, C. L. (2008). Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems. *Applied Mathematics and Computation*, 205(2), 550-561.
- Chang, P. C., Hsieh, J. C., & Lin, S. G. (2002). The development of gradual-priority weighting approach for the multi-objective flowshop scheduling problem. *International Journal of Production Economics*, 79(3), 171-183.
- Chen, C. L., Vempati, V. S., & Aljaber, N. (1995). An application of genetic algorithms for flow shop problems. *European Journal of Operational Research*, 80(2), 389-396.

- Çolak, S. (2010). Genetik Algoritmalar Yardımı ile Gezgin Satıcı Probleminin Çözümü Üzerine Bir Uygulama. *Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 19(3).
- Das, S. R., Gupta, J. N., & Khumawala, B. M. (1995). A savings index heuristic algorithm for flowshop scheduling with sequence dependent set-up times. *Journal of the Operational Research Society*, 46(11), 1365-1373.
- Datta, S., Sahu, N., & Mahapatra, S. (2013). Robot selection based on grey-MULTIMOORA approach. *Grey Systems: Theory and Application*, 3(2), 201-232.
- Dilaver, D. (2015). Genetik Algoritmalar Yardımıyla İş Atölye Çizelgelemesi Üzerine Bir Çalışma. Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü.
- Dugardin, F., Yalaoui, F., & Amodeo, L. (2010). New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research*, 203(1), 22-31.
- Eren, T., & Güner, E. (2005). İki Ölçütlü Beklemesiz Akış Tipi Çizelgeleme Problemi: Toplam Tamamlanma Zamanı ve Maksimum Gecikme. V. Ulusal Üretim Araştırmaları Sempozyumu.
- Gerşil, M., & Palamutçuoğlu, T. (2013). Ders çizelgeleme probleminin melez genetik algoritmalar ile performans analizi. *Ömer Halisdemir Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 6(1), 242.
- Gonçalves, J. F., de Magalhães Mendes, J. J., & Resende, M. G. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European journal of operational research*, 167(1), 77-95.
- Gözen, Ş. (2007). Bulanık esnek akış tipi çok prosesli çizelgeleme problemlerinin genetik algoritma ve tavlama benzetimi ile çözümü. Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü.
- Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 82-87.
- Hosseini, S. M. H. (2017). A multi-objective genetic algorithm (MOGA) for hybrid flow shop scheduling problem with assembly operation. *Journal of Industrial and Systems Engineering*, 10, 132-154.
- Ishibuchi, H., & Murata, T. (1996). Multi-objective genetic local search algorithm. In *Evolutionary Computation, Proceedings of IEEE International Conference on*, 119-124.
- Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3), 392-403.

- Ishibuchi, H., Yoshida, T., & Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE transactions on evolutionary computation*, 7(2), 204-223.
- Johnson, S. M. (1954). Optimal two- and three- stage production schedules with setup times included. *Naval Research Logistics (NRL)*, 1(1), 61-68.
- Kamışlı Öztürk, Z., & Sağır Özdemir, M. (2005). Çok ölçütlü çizelgeleme problemleri: gelinen nokta ve potansiyel araştırma alanları. V. Ulusal Üretim Araştırmaları Sempozyumu.
- Keskin, K. (2010). Beklemesiz akış tipi çizelgeleme problemlerinin çok amaçlı melez genetik algoritma ile çözümü. *Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Li, B. B., & Wang, L. (2007). A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(3), 576-591.
- Li, B. B., Wang, L., & Liu, B. (2008). An effective PSO-based hybrid algorithm for multiobjective permutation flow shop scheduling. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, 38(4), 818-831.
- Lin, D., Lee, C. K. M., & Wu, Z. (2011). Integrated GA and AHP for re-entrant flow shop scheduling problem. *International Conference on Quality and Reliability*, 496-500.
- Lin, D., Lee, C. K. M., & Wu, Z. (2012). Integrating analytical hierarchy process to genetic algorithm for re-entrant flow shop scheduling problem. *International Journal of Production Research*, 50(7), 1813-1824.
- Majazi Dalfard, V., Ardakani, A., & Nazalsadat Banihashemi, T. (2011). Hybrid genetic algorithm for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times. *Tehnički vjesnik*, 18(4), 497-504.
- Marichelvam, M. K., Prabakaran, T., & Yang, X. S. (2014). A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE transactions on evolutionary computation*, 18(2), 301-305.
- Mokhtari, H., Abadi, I. N. K., & Cheraghalikhani, A. (2011). A multi-objective flow shop scheduling with resource-dependent processing times: trade-off between makespan and cost of resources. *International Journal of Production Research*, 49(19), 5851-5875.
- Murata, T., Ishibuchi, H., & Tanaka, H. (1996a). Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering*, 30(4), 1061-1071.
- Murata, T., Ishibuchi, H., & Tanaka, H. (1996b). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering*, 30(4), 957-968.
- Pan, Q. K., Wang, L., & Qian, B. (2009). A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems. *Computers & Operations Research*, 36(8), 2498-2511

- Pasupathy, T., Rajendran, C., & Suresh, R. K. (2006). A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs. *The International Journal of Advanced Manufacturing Technology*, 27(7), 804-815.
- Ponnambalam, S. G., Jagannathan, H., Kataria, M., & Gadicherla, A. (2004). A TSP-GA multi-objective algorithm for flow-shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 23(11-12), 909-915.
- Poon, P. W., & Carter, J. N. (1995). Genetic algorithm crossover operators for ordering applications. *Computers & Operations Research*, 22(1), 135-147.
- Pour, N., Tavakkoli-Moghaddam, R., & Asadi, H. (2013). 5. Optimizing a multi-objectives flow shop scheduling problem by a novel genetic algorithm. *International Journal of Industrial Engineering Computations*, 4(3), 345-354.
- Rahimi-Vahed, A. R., & Mirghorbani, S. M. (2007). A multi-objective particle swarm for a flow shop scheduling problem. *Journal of combinatorial optimization*, 13(1), 79-102.
- Rahimi-Vahed, A., Dangchi, M., Rafiei, H., & Salimi, E. (2009). A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 41(11), 1227-1239.
- Rajendran, C., & Chaudhuri, D. (1990). Heuristic algorithms for continuous flow-shop problem. *Naval Research Logistics (NRL)*, 37(5), 695-705.
- Rashidi, E., Jahandar, M., & Zandieh, M. (2010). An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. *The International Journal of Advanced Manufacturing Technology*, 49(9), 1129-1139.
- Şevkli, M. (2005). Atölye Tipi Çizelgeleme Problemlerine Parçacık Sürü Optimizasyonu Yaklaşımı ve Genetik Algoritma Modeli ile Karşılaştırılması. Doktora Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., & Mirzaei, A. H. (2007). A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. *Information Sciences*, 177(22), 5072-5090.
- Temiz, İ. (2010). Çok Kriterli Permütasyon Akış Tipi Çizelgeleme Problemi için Bir Tavlama Benzetimi Yaklaşımı. Cankaya University Journal of Science and Engineering, 7(2).
- Temiz, İ., & Erol, S. (2007). Bulanık Akış Tipi Çizelgeleme Problemi İçin Çok Amaçlı Genetik Algoritma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 22(4).
- Türe, H., Koçak, D., & Doğan, S. (2017). Multimoora Yöntemi İle Ülke Riski Değerlendirmesi. *İktisadi Ve İdari Bilimler Fakültesi Dergisi*, 18(3), 824-844.

- Veeraiah, T., Pratapa Reddy, Y., V S Mohan Kumar, P., & W D S Milton, P. (2017). Optimization of Flow Shop Scheduling by MATLAB. *SSRG International Journal of Mechanical Engineering (SSRG-IJME) – Special Issue May – 2017*, 222-226.
- Wu, Y., Liu, J., & Peng, C. (2014). A New Replacement Strategy for Genetic Algorithm and Computational Experiments. In *Computer, Consumer and Control (IS3C), 2014 International Symposium on* (pp. 733-736). IEEE.
- Xie, H., Zhang, M., Andreae, P., & Johnston, M. (2008, June). Is the not-sampled issue in tournament selection critical?. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on* (pp. 3710-3717). IEEE.
- Yağmahan, B., & Yenisey, M. M. (2006). Akış tipi çizelgeleme problemi için KKE parametre eniyileme. *İTÜDERGİSİ/d*, 5(2), 133-141.
- Zandieh, M., & Karimi, N. (2011). An adaptive multi-population genetic algorithm to solve the multi-objective group scheduling problem in hybrid flexible flowshop with sequence-dependent setup times. *Journal of Intelligent Manufacturing*, 22(6), 979-989.

EKLER

EK 1: 15 işten oluşan akış tipi çizelgeleme problemi için işlerin teslim süreleri ve tezgahlardaki işlem süreleri tablosu

İş No	Teslim Süresi (Dakika)	Tezgah No									
		1	2	3	4	5	6	7	8	9	10
		İşlem Süreleri (Dakika)									
1	350	16	18	12	10	8	12	8	10	16	18
2	490	14	10	11	16	32	11	32	16	14	10
3	540	13	20	15	12	22	15	22	12	13	20
4	400	19	15	19	21	13	19	13	21	19	15
5	324	15	16	16	10	12	16	12	10	15	16
6	290	15	13	11	22	13	11	13	22	15	13
7	350	20	22	15	20	19	15	19	20	20	22
8	545	13	20	24	11	14	19	14	11	13	20
9	255	8	18	11	15	20	11	24	15	8	18
10	500	20	15	24	19	24	18	24	23	19	15
11	260	12	8	10	16	18	16	18	12	10	8
12	465	11	32	16	14	10	14	10	11	16	32
13	415	15	22	12	13	20	13	20	15	12	22
14	360	19	13	21	19	15	19	15	19	21	13
15	420	16	12	10	15	16	15	16	16	10	12

EK 2: 25 işten oluşan akış tipi çizelgeleme problemi için işlerin teslim süreleri ve tezgahlardaki işlem süreleri tablosu

İş No	Teslim Süresi (Dakika)	Tezgah No									
		1	2	3	4	5	6	7	8	9	10
		İşlem Süreleri (Dakika)									
1	350	16	18	12	10	8	12	8	10	16	18
2	490	14	10	11	16	32	11	32	16	14	10
3	540	13	20	15	12	22	15	22	12	13	20
4	400	19	15	19	21	13	19	13	21	19	15
5	324	15	16	16	10	12	16	12	10	15	16
6	290	15	13	11	22	13	11	13	22	15	13
7	350	20	22	15	20	19	15	19	20	20	22
8	545	13	20	24	11	14	19	14	11	13	20
9	255	8	18	11	15	20	11	24	15	8	18
10	500	20	15	24	19	24	18	24	23	19	15
11	260	12	8	10	16	18	16	18	12	10	8
12	465	11	32	16	14	10	14	10	11	16	32
13	415	15	22	12	13	20	13	20	15	12	22
14	360	19	13	21	19	15	19	15	19	21	13
15	420	16	12	10	15	16	15	16	16	10	12
16	440	11	13	22	15	13	15	13	11	22	13
17	270	15	19	20	20	22	20	22	15	20	19
18	520	24	14	11	13	20	13	20	24	11	14
19	320	11	24	15	8	18	8	18	11	15	24
20	495	15	24	23	14	15	20	15	24	23	19
21	420	15	16	24	15	19	18	20	16	12	15
22	380	20	18	16	15	19	14	17	11	14	10
23	290	16	11	9	12	18	16	13	15	17	21
24	360	8	16	13	11	19	21	15	16	19	14
25	440	16	12	19	17	10	16	13	15	19	14

EK 3: 50 işten oluşan akış tipi çizelgeleme problemi için işlerin teslim süreleri ve tezgahlardaki işlem süreleri tablosu

İş No	Teslim Süresi (Dakika)	Tezgah No									
		1	2	3	4	5	6	7	8	9	10
		İşlem Süreleri (Dakika)									
1	350	16	18	12	10	8	12	8	10	16	18
2	490	14	10	11	16	32	11	32	16	14	10
3	540	13	20	15	12	22	15	22	12	13	20
4	400	19	15	19	21	13	19	13	21	19	15
5	324	15	16	16	10	12	16	12	10	15	16
6	290	15	13	11	22	13	11	13	22	15	13
7	350	20	22	15	20	19	15	19	20	20	22
8	545	13	20	24	11	14	19	14	11	13	20
9	255	8	18	11	15	20	11	24	15	8	18
10	500	20	15	24	19	24	18	24	23	19	15
11	260	12	8	10	16	18	16	18	12	10	8
12	465	11	32	16	14	10	14	10	11	16	32
13	415	15	22	12	13	20	13	20	15	12	22
14	360	19	13	21	19	15	19	15	19	21	13
15	420	16	12	10	15	16	15	16	16	10	12
16	440	11	13	22	15	13	15	13	11	22	13
17	270	15	19	20	20	22	20	22	15	20	19
18	520	24	14	11	13	20	13	20	24	11	14
19	320	11	24	15	8	18	8	18	11	15	24
20	495	15	24	23	14	15	20	15	24	23	19
21	420	15	16	24	15	19	18	20	16	12	15
22	380	20	18	16	15	19	14	17	11	14	10
23	290	16	11	9	12	18	16	13	15	17	21
24	360	8	16	13	11	19	21	15	16	19	14
25	440	16	12	19	17	10	16	13	15	19	14
26	320	13	15	16	14	19	18	20	16	14	18
27	230	12	16	14	15	17	20	19	16	13	15
28	450	9	12	16	14	13	17	14	19	21	25
29	380	14	20	19	23	18	15	16	21	9	11
30	310	16	17	19	16	17	15	11	13	8	12
31	420	12	9	10	13	8	11	18	16	14	12
32	450	11	13	9	14	16	20	18	12	10	16

33	390	9	11	15	19	21	17	15	16	10	11
34	510	17	19	20	15	11	16	13	18	14	19
35	480	14	16	20	21	18	14	16	19	15	13
36	450	12	10	16	20	18	17	9	11	13	14
37	480	13	15	19	17	14	13	11	12	12	15
38	390	9	10	18	11	13	14	12	11	10	14
39	520	11	17	21	20	15	19	14	16	17	12
40	500	17	15	16	14	19	17	16	14	9	14
41	490	18	12	14	16	17	14	11	9	16	12
42	460	10	16	18	17	14	12	13	11	15	16
43	470	18	11	16	17	13	12	11	12	14	17
44	370	19	10	9	10	11	10	17	16	13	15
45	450	14	17	16	19	11	12	14	17	20	19
46	470	17	16	14	12	19	18	13	12	11	13
47	390	14	16	19	12	21	20	18	17	16	19
48	380	8	12	9	14	16	12	11	17	14	13
49	410	10	16	19	17	18	14	12	11	10	16
50	450	14	19	14	16	10	9	11	14	16	18

EK 4: 100 işten oluşan akış tipi çizelgeleme problemi için işlerin teslim süreleri ve tezgahlardaki işlem süreleri tablosu

İş No	Teslim Süresi (Dakika)	Tezgah No									
		1	2	3	4	5	6	7	8	9	10
		İşlem Süreleri (Dakika)									
1	350	16	18	12	10	8	12	8	10	16	18
2	490	14	10	11	16	32	11	32	16	14	10
3	540	13	20	15	12	22	15	22	12	13	20
4	400	19	15	19	21	13	19	13	21	19	15
5	324	15	16	16	10	12	16	12	10	15	16
6	290	15	13	11	22	13	11	13	22	15	13
7	350	20	22	15	20	19	15	19	20	20	22
8	545	13	20	24	11	14	19	14	11	13	20
9	255	8	18	11	15	20	11	24	15	8	18
10	500	20	15	24	19	24	18	24	23	19	15
11	260	12	8	10	16	18	16	18	12	10	8
12	465	11	32	16	14	10	14	10	11	16	32
13	415	15	22	12	13	20	13	20	15	12	22
14	360	19	13	21	19	15	19	15	19	21	13
15	420	16	12	10	15	16	15	16	16	10	12
16	440	11	13	22	15	13	15	13	11	22	13
17	270	15	19	20	20	22	20	22	15	20	19
18	520	24	14	11	13	20	13	20	24	11	14
19	320	11	24	15	8	18	8	18	11	15	24
20	495	15	24	23	14	15	20	15	24	23	19
21	420	15	16	24	15	19	18	20	16	12	15
22	380	20	18	16	15	19	14	17	11	14	10
23	290	16	11	9	12	18	16	13	15	17	21
24	360	8	16	13	11	19	21	15	16	19	14
25	440	16	12	19	17	10	16	13	15	19	14
26	320	13	15	16	14	19	18	20	16	14	18
27	230	12	16	14	15	17	20	19	16	13	15
28	450	9	12	16	14	13	17	14	19	21	25
29	380	14	20	19	23	18	15	16	21	9	11
30	310	16	17	19	16	17	15	11	13	8	12
31	420	12	9	10	13	8	11	18	16	14	12
32	450	11	13	9	14	16	20	18	12	10	16
33	390	9	11	15	19	21	17	15	16	10	11
34	510	17	19	20	15	11	16	13	18	14	19
35	480	14	16	20	21	18	14	16	19	15	13
36	450	12	10	16	20	18	17	9	11	13	14

37	480	13	15	19	17	14	13	11	12	12	15
38	390	9	10	18	11	13	14	12	11	10	14
39	520	11	17	21	20	15	19	14	16	17	12
40	500	17	15	16	14	19	17	16	14	9	14
41	490	18	12	14	16	17	14	11	9	16	12
42	460	10	16	18	17	14	12	13	11	15	16
43	470	18	11	16	17	13	12	11	12	14	17
44	370	19	10	9	10	11	10	17	16	13	15
45	450	14	17	16	19	11	12	14	17	20	19
46	470	17	16	14	12	19	18	13	12	11	13
47	390	14	16	19	12	21	20	18	17	16	19
48	380	8	12	9	14	16	12	11	17	14	13
49	410	10	16	19	17	18	14	12	11	10	16
50	450	14	19	14	16	10	9	11	14	16	18
51	400	16	18	12	18	8	12	11	10	12	18
52	490	14	10	11	16	32	11	32	16	14	10
53	520	13	19	15	12	15	15	22	12	13	20
54	400	19	15	17	21	15	19	13	21	20	15
55	340	15	16	12	10	12	16	15	10	15	19
56	300	15	13	11	18	13	11	17	22	15	13
57	380	20	11	12	20	19	15	14	20	13	22
58	510	13	20	19	11	14	20	14	12	13	18
59	255	8	18	11	15	24	11	24	15	8	18
60	420	24	15	19	23	17	11	24	23	15	15
61	290	12	8	10	12	11	14	18	12	10	8
62	440	11	20	16	14	10	14	10	11	16	10
63	415	15	22	12	13	20	13	20	15	12	22
64	360	19	13	21	19	15	19	15	19	21	13
65	420	15	12	10	15	16	11	16	17	10	12
66	440	11	13	22	15	13	15	13	11	22	13
67	270	15	19	20	20	22	20	22	15	20	19
68	520	24	14	11	13	20	13	20	24	11	14
69	240	11	19	15	8	18	8	18	11	15	20
70	495	12	24	23	10	15	18	15	24	19	24
71	420	15	16	24	15	24	18	20	16	12	15
72	360	15	18	16	15	19	14	17	11	14	10
73	300	16	11	9	12	18	16	13	15	17	21
74	290	8	16	13	11	19	21	15	16	19	14
75	440	16	12	19	17	10	16	13	15	19	14
76	320	13	15	16	14	19	18	20	16	14	18
77	230	12	16	14	15	17	20	19	16	13	15
78	450	9	12	16	14	13	17	14	20	21	25
79	380	14	20	17	23	18	16	16	21	9	11

80	310	16	17	19	16	17	15	11	13	8	12
81	450	15	9	10	13	8	11	18	16	14	12
82	430	11	13	9	14	16	20	18	12	10	16
83	390	9	11	15	19	21	17	15	16	10	11
84	510	17	19	20	15	11	16	13	18	14	19
85	480	14	16	20	21	18	14	16	19	15	13
86	450	12	10	16	20	18	17	9	11	13	14
87	480	16	15	19	17	14	13	15	11	12	15
88	390	9	10	18	9	13	14	12	11	10	14
89	520	11	17	21	20	15	19	24	16	17	12
90	500	17	15	16	14	19	17	16	14	9	14
91	490	18	12	14	16	17	14	11	9	16	12
92	470	10	16	18	17	14	12	13	11	15	16
93	460	18	11	16	15	13	12	11	19	14	17
94	370	19	10	9	10	11	10	17	16	13	15
95	450	14	17	16	19	11	12	14	17	20	19
96	470	17	16	14	12	19	18	13	12	11	13
97	420	14	16	20	12	21	20	18	18	16	19
98	380	8	12	9	14	16	12	11	17	14	13
99	410	10	16	19	17	18	14	12	12	10	16
100	450	16	19	14	16	10	9	11	14	16	18

ÖZGEÇMİŞ

Mine Büşra Gelen, 24.03.1992’de Kocaeli’de doğdu. İlk, orta ve lise öğrenimini Kocaeli’de tamamladı. 2011 yılında başladığı Sakarya Üniversitesi Endüstri Mühendisliği Bölümü’ndeki eğitimini 2015 yılında tamamladı. 2015 yılında Anadolu Üniversitesi Lojistik Programı’ndaki eğitimine başladı. 2016 yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü’nde yüksek lisans eğitimine başladı. 2017 yılında Lojistik Programı’ndaki eğitimini tamamladı ve ardından Sakarya Üniversitesi İşletme Fakültesi’nde Uluslararası Ticaret Bölümü lisans eğitimine başladı. Halen Sakarya Üniversitesi Endüstri Mühendisliği Bölümü yüksek lisans ve Uluslararası Ticaret Bölümü lisans eğitimine devam etmektedir.